



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA STROJNÍHO INŽENÝRSTVÍ
ÚSTAV AUTOMATIZACE A INFORMATIKY

FACULTY OF MECHANICAL ENGINEERING
INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

LABORATORNÍ ÚLOHA ŘÍZENÍ POHYBU PŘI PŘEMISŤOVÁNÍ ZAVĚŠENÉHO PŘEDMĚTU

LABORATORY TASK OF MOTION CONTROL DURING HANGET OBJECT TRANSPORTATION

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. ONDŘEJ PRAŽÁK

VEDOUCÍ PRÁCE
SUPERVISOR

doc. Ing. ZDENĚK NĚMEC, CSc.

BRNO 2013

Vysoké učení technické v Brně, Fakulta strojního inženýrství

Ústav automatizace a informatiky

Akademický rok: 2012/2013

ZADÁNÍ DIPLOMOVÉ PRÁCE

student(ka): Bc. Ondřej Pražák

který/která studuje v **magisterském navazujícím studijním programu**

obor: **Aplikovaná informatika a řízení (3902T001)**

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

Laboratorní úloha řízení pohybu při přemísťování zavěšeného předmětu

v anglickém jazyce:

Laboratory task of motion control during hanget object transportation

Stručná charakteristika problematiky úkolu:

Jde o laboratorní ukázkou řízení silně kmitající soustavy v souvislosti s přemísťováním zavěšených předmětů. Vyžaduje se funkční vzor s daným přístrojovým uspořádáním a vybavením.

Cíle diplomové práce:

1. Upravit fyzikální model z práce [1] na zkoušení pohybu zavěšeného předmětu.
3. Navrhnout varianty řešení a algoritmů pro řízení pohybu při přemísťování předmětu.
4. Zpracovat uživatelské programové vybavení pro řídicí automat a operátorský panel.
5. Navrhnout ukázky laboratorních cvičení s dotyčnou úlohou.

Seznam odborné literatury:

- [1] Vrtal,V.: Laboratorní ukázka řízení asynchronního motoru. Diplomová práce VUT v Brně, FSI-ÚAI,2010.
- [2] Dokumentace fy Siemens k měniči Sinamics G 110, k automatu Simatic S7-224XP, k panelu Simatic TP 177 micro.
- [3] Dokumentace programového vybavení Step7-Micro/Win a WinCC flexible Micro.

Vedoucí diplomové práce: doc. Ing. Zdeněk Němec, CSc.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2012/2013.

V Brně, dne 20.11.2012

L.S.

Ing. Jan Roupec, Ph.D.
Ředitel ústavu

prof. RNDr. Miroslav Doupovec, CSc., dr. h. c.
Děkan fakulty

Abstrakt

Diplomová práce se zabývá automatickým řízením pro přemísťování zavěšených předmětů. Jedná se o regulační úlohu se silně kmitavou soustavu, pro kterou byl zhotoven i fyzikální zkušební model. Pro tento model byla navržena aplikace pro řízení pomocí programovatelného automatu, frekvenčního měniče a operátorského panelu, vše od firmy Siemens.

Abstract

The diploma thesis deals with automatic control of transferring suspended objects. It is a strongly oscillating system for which a physical trial model has been made. An application has been made for this model, which controls this system by means of PLC, frequency inverter and operating panel, everything by Siemens.

Klíčová slova

Kmitavá soustava, zavěšený předmět, programovatelný automat, operátorský panel, přenos regulátoru, přenos soustavy, frekvenční měnič

Keywords

Oscillating systems, suspended object, a PLC, operator panel, transmission of regulator, transmission of system, frequency driver

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci na téma: „Laboratorní úloha řízení pohybu při přemísťování zavěšeného předmětu.“, vypracoval samostatně pod vedením doc. Ing. Zdeňka Němce, CSc., na základě dostupné literatury a dostupných informačních zdrojů, které jsem všechny odcitoval v seznamu použité literatury.

V Brně dne

podpis.....

Bibliografická citace mé práce:

PRAŽÁK, O. *Laboratorní úloha řízení pohybu při přemísťování zavěšeného předmětu*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2013. 60s. Vedoucí diplomové práce doc. Ing. Zdeněk Němec, CSc.

Poděkování

Velký dík patří vedoucímu mé diplomové práce doc. Ing. Zdeňku Němcovi, CSc.. Rád bych mu tímto poděkoval za cenné rady, podněty a odborné vedení, při vypracování této diplomové práce.

Obsah

1	Úvod.....	11
2	Popis součástí	12
2.1	Programovatelný automat SIMATIC S7 224 XP.....	12
2.1.1	Základní popis PLC SIMATIC S7 224 XP	12
2.1.2	Technické parametry PLC SIMATIC S7 224 XP	12
2.1.3	Propojení frekvenčního měniče s automatem	13
2.1.4	Vytvoření programu pomocí Step 7-MicroWIN	14
2.2	Operátorský panel SIMATIC TP 177 micro	16
2.2.1	Základní popis operátorského panelu SIMATIC TP 177 micro	16
2.2.2	Technické parametry operátorského panelu SIMATIC TP 177 micro	17
2.2.3	Vytvoření programu pomocí SIMATIC WinCC flexible 2008 Micro	17
2.3	Frekvenční měnič SINAMICS G110	19
2.3.1	Základní popis frekvenčního měniče SINAMICS G110	19
2.3.2	Technické parametry měniče SINAMICS G110.....	20
2.3.3	Blokové schéma měniče a jeho popis.....	20
2.3.4	Popis knihovny USS.....	22
2.4	Asynchronní motor s impulzním snímačem otáček	23
2.4.1	Základní popis asynchronního motoru	23
2.4.2	Technické specifikace třífázového asynchronního motoru SIEMENS	24
2.4.3	Impulzní snímač otáček.....	25
2.5	Odměřování polohy tělesa.....	25
3	Popis celé laboratorní soustavy	27
3.1	Blokové schéma laboratorní soustavy a její popis	27
3.2	Schéma mechanismu kočky a její popis.....	29
4	Návrh regulace polohy zavěšeného tělesa.....	31

4.1	První varianta řízení	31
4.1.1	Nastavení regulátoru.....	32
4.2	Druhá varianta řízení	33
4.2.1	Měření a experimenty	34
5	Pokročilé formy řízení.....	38
5.1	Stavová regulace.....	38
5.1.1	Stavový popis soustavy	38
5.2	Prediktivní regulace.....	40
6	Řešení softwarového vybavení automatu	41
6.1	Vývojový diagram programového vybavení automatu	41
6.2	Podrobný popis programu	43
6.2.1	Hlavní program Main	43
6.2.2	Podprogram počáteční nastavení	45
6.2.3	Podprogram přerušení pro rychlostní smyčku regulátoru	46
6.2.4	Podprogram přerušení pro polohovou smyčku regulátoru	48
7	Navržené uživatelské ovládací prostředí	50
7.1	Struktura uživatelského prostředí.....	50
7.2	Popis a ukázky jednotlivých obrazovek uživatelského prostředí.....	51
8	Laboratorní úloha řízení pohybu při přemísťování zavěšeného předmětu	55
9	Závěr	57
	Použitá literatura.....	58

1 Úvod

Řízení zavěšených předmětů se v současnosti využívá v mnoha oborech. Nejčastěji jako řízení jeřábových koček při přemísťování břemen jeřáby. V dnešní době je mnoho různých typů a konstrukcí jeřábů. Proto jsou rozděleny do několika tříd. Jsou to zdvihadla, to jsou spíše ruční a malé jeřáby, kterých se tato problematika příliš netýká. Dále pak věžové, sloupové, mobilní a speciální jeřáby. Mezi speciální se řadí deriky, lanové a portálové jeřáby.

Na všechny tyto zařízení působí mnoho nepříznivých vlivů, hlavně setrvačná síla, působící na těleso zavěšené na laně, povětrnostní vlivy, atd. Díky vychýlení tělesa z jeho rovnovážného stavu se lano nemusí vždy navíjet pod správným úhlem a může docházet k většímu opotřebením lan. Také vlivem nesprávné manipulace a nezkušeností obsluhy se může těleso dostat mimo předpokládanou polohu a může způsobit škody. Proto se mnoho firem snaží vyvinout takové zařízení, které by bylo alespoň částečně schopno tyto vlivy odstranit.

Mým úkolem bylo vytvořit k danému hardwaru takové řízení, které by nepříznivé vlivy při přemísťování odstranilo. Tedy měřit hodnotu vychýlení zavěšeného tělesa, v tomto případě vychýlení v jedné ose, a vytvořit různé metody regulace pomocí programovatelného automatu SIMATIC S7 224XP, které by toto vychýlení eliminovaly. Poloha tělesa je odměřována potenciometrem, který je součástí jeřábové kočky.

Dále bylo vytvořeno uživatelské prostředí v operátorském panelu, kterým bude moci uživatel ovládat přemísťování zavěšeného předmětu. Nevýhodou této soustavy je neschopnost pamatovat si svou polohu při odpojení ze sítě, proto je při spuštění celé soustavy nutno nastavit výchozí polohu. Žádost o provedení je uživateli sdělena přes operátorský panel. Při splnění těchto podmínek se uživatel dostane k ovládacím prvkům a může soustavu ovládat. Na závěr shrnu poznatky z této práce.

2 Popis součástí

2.1 Programovatelný automat SIMATIC S7 224 XP

Tato kapitola vychází z [1]

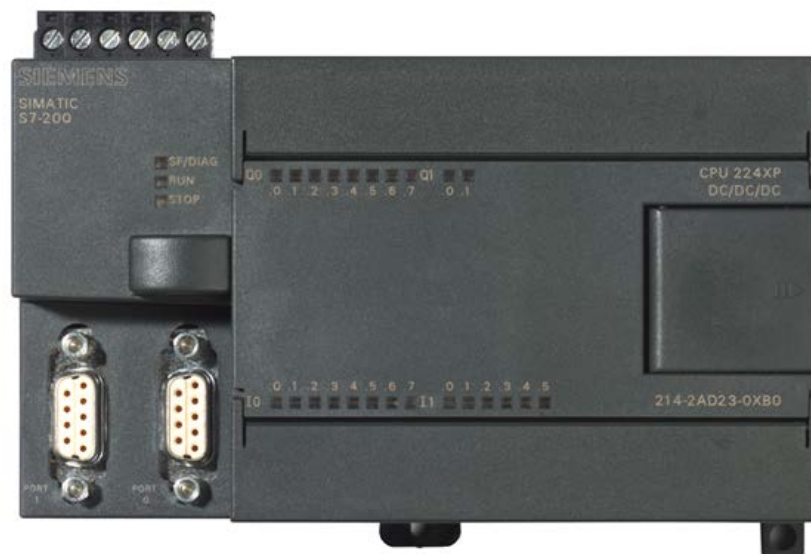
2.1.1 Základní popis PLC SIMATIC S7 224 XP

SIMATIC S7 224 XP patří mezi malé programovatelné automaty, takzvané mikro-PLC. Vynikají svým kompaktním designem, vysokým výkonem při řízení v reálném čase a velkou programovou pamětí. Tento automat je v základu vybaven 14 digitálními a 2 analogovými vstupy, 10 digitálními a jedním analogovým výstupem a také dvěma porty RS485 pro komunikaci. V mém případě je jeden použit pro komunikaci s operátorským panelem a druhý pro komunikaci s frekvenčním měničem. Přes tyto porty je zároveň možno nahrávat program do automatu a operátorského panelu. Přidáním dalších modulů k tomuto automatu je možné zvyšovat počty digitálních i analogových vstupů a výstupů. Jejich maximální počet, ale i ostatní parametry jsou vypsány v kapitole 2.1.2. Dále SIMATIC S7 224 XP obsahuje 6 vysokorychlostních čítačů, z toho dva jsou schopny pracovat s frekvencí až 200kHz, a dva pulzní výstupy s frekvencí až 100 kHz.

2.1.2 Technické parametry PLC SIMATIC S7 224 XP

Integrované digitální vstupy/výstupy	14DI/10DO
Digitální vstupy/výstupy/maximum	92/82/168
Integrované analogové vstupy/výstupy	2/1
Analogové vstupy/výstupy/maximum	30/15/38
Paměť pro program/data	12KB/10KB
Vysokorychlostní čítače	6 celkem, 2x200kHz
Komunikační porty RS485	2
Integrovaný zdroj 24V DC	max. 280mA

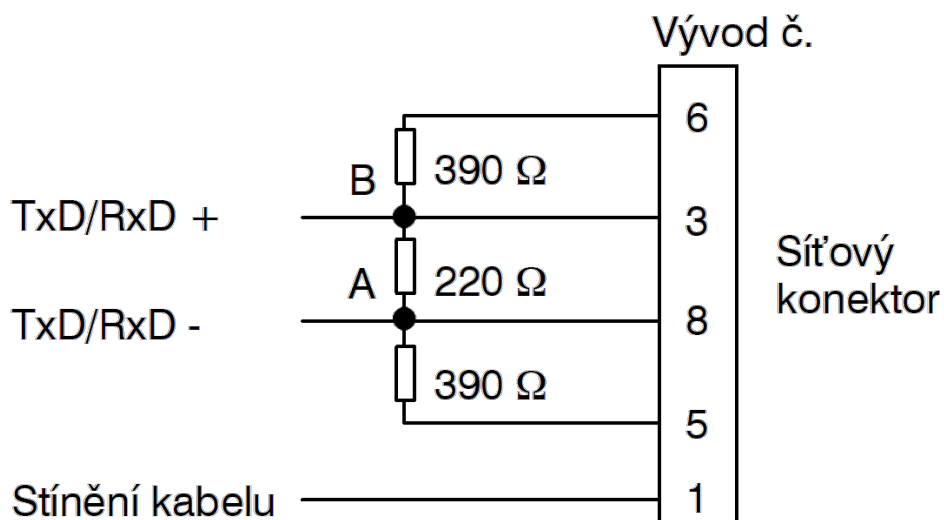
Tabulka 1 Technické parametry PLC SINAMISC S7 224 XP [1]



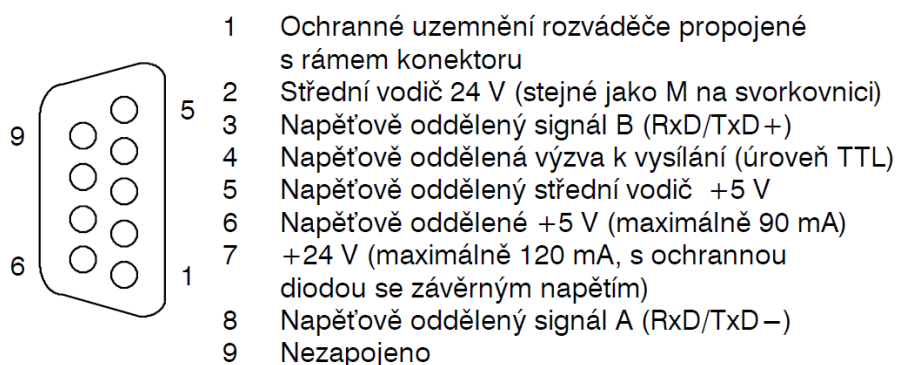
Obr. 2.1 PLC SIMATICS S7 224 XP [2]

2.1.3 Propojení frekvenčního měniče s automatem

„Pro propojení automatu s frekvenčním měničem bylo zapotřebí zhotovit kabel, protože nebyl součástí setu. Jednalo se o běžný 9-pinový CAN konektor typu samec. Koncová zařízení na sběrnici RS 485 musí být zakončena odporovou zátěží. Zakončení na straně měniče se provádí pomocí přepínače a na straně automatu se zakončení provádí zapojením konektoru podle Obr. 2.2. Takto zakončená sběrnice může měřit až 1200 m. V našem případě, kdy máme propojená jenom dvě zařízení a délka kabeláže je jen několik desítek centimetrů, můžeme zakončovací odpory vynechat. Vynechání zakončovacích odporů na straně automatu nemá vliv na funkci sběrnice (v našem případě).“ [3]



Obr. 2.2 Zapojení RS 485 se zakončovacími odpory [3]



Obr. 2.3 Popis jednotlivých pinů konektoru RS 485 [1]

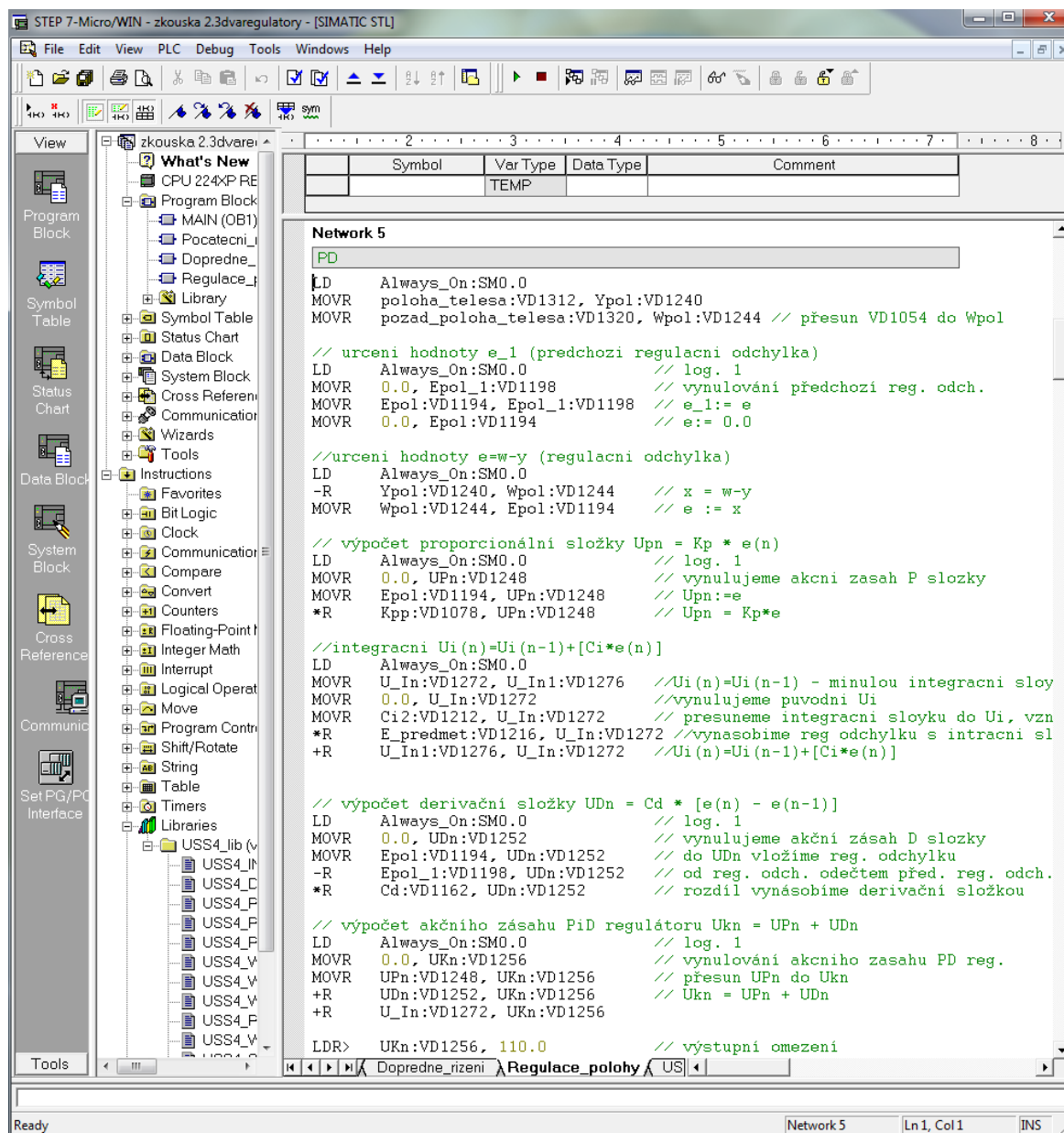
2.1.4 Vytvoření programu pomocí Step 7-MicroWIN

Step 7-MicroWIN (Obr. 2.4) je uživatelské prostředí na tvorbu programů pro programovatelné automaty. V horní části je rozložení nabídky podobné jako ve většině běžných programů. Je zde lišta se základními nástroji jako například tisk, uložení či otevření programu. Vedlejší lišta nabízí nástroje pro kompilaci, sledování běhu programu nebo tlačítka pro nahrávání programu do automatu či z automatu. Stejně jako u jiných programů lze tyto lišty skrýt, popřípadě zobrazit jiné s jinými nástroji.

Dále se pak v levé části okna nachází lišta pro navigaci. Vedle strom instrukcí a v pravé části je samotný programový editor. V liště pro navigaci jsou ikony pro zobrazení programového bloku, tabulky symbolů, datového a systémového bloku a také ikona pro nastavení komunikace automatu s ostatními zařízeními.

Strom instrukcí skrývá všechny části programu jako například MAIN, jednotlivé části programu, programy přerušení a v mém případě i programové bloky protokolu USS. Níže jsou veškeré instrukce, které lze vkládat dvojím kliknutím na danou instrukci či přetažením do programového editoru. Na konci jsou pak knihovny, jako například v mém případě knihovny USS.

Programový editor nabízí tři druhy programování. Programování v kontaktním schématu (LAD), pomocí výpisu příkazů (STL) nebo funkčními bloky (FBD). Editory LAD a FBD mají určitá omezení při programování, proto je nejčastěji používán výpis příkazů STL který jsem využíval i já.



Obr. 2.4 Uživatelské prostředí Step 7-MicroWIN

Editor STL a jeho popis [1]

Editor STL zobrazuje program jako znakově orientovaný programovací jazyk. Umožňuje vytvářet řídicí programy vkládáním textových instrukcí. Editor STL také umožňuje tvorbu programů, které by pomocí editorů LAD nebo FBD nešly vytvořit. Je to proto, že v STL programujete v jazyku S7-200, a nikoli v jazyku grafického editoru, kde platí určitá omezení, aby byly diagramy správně nakresleny. Jak je vidět na Obr. 2.5, je tato znakově orientovaná koncepce velmi podobná programování ve strojovém kódu. S7-200 provádí každou instrukci v pořadí určeném programem shora dolů, pak začne opět odshora. Pro řešení řídicí logiky používá STL logický zásobník.

```

MOVR   Pocet_skut_ot:VD1050, poloha_kocky:VD1316 //přepočet skutečných t
*R      5.0, poloha_kocky:VD1316

MOVR   poloha_kocky:VD1316, poloha_telesa:VD1312 //přičtení polohy těles
+R      pol_předm_uhlova:VD1308, poloha_telesa:VD1312

```

Obr. 2.5 Ukázka kódu vytvořeného v editoru STL

Charakteristiky editoru STL:

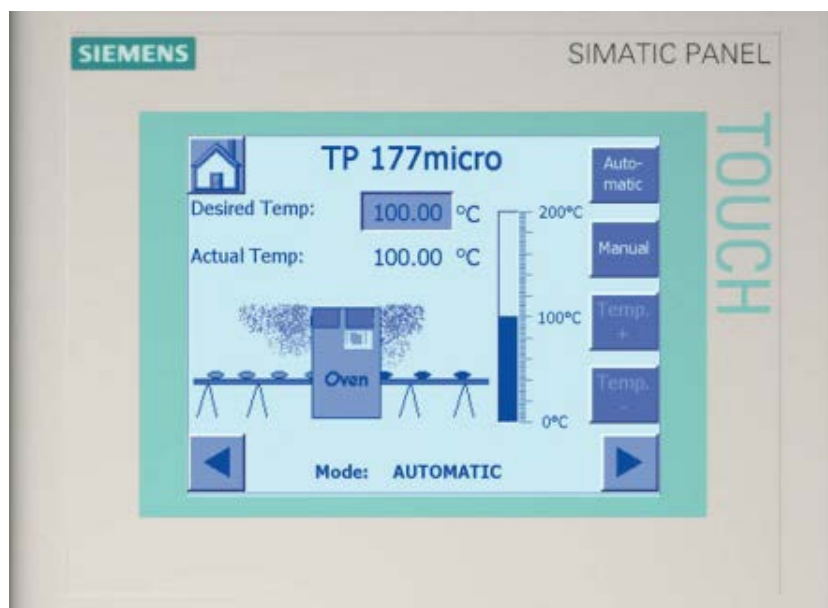
- STL je nejvhodnější pro zkušené programátory.
- STL někdy umožní řešit problémy, které se nedají snadno řešit pomocí editorů LAD a FBD.
- Editor STL můžete používat pouze s instrukčním souborem SIMATIC.
- Editor STL můžete vždy použít na prohlížení nebo editaci programu, který byl vytvořen pomocí editorů LAD nebo FBD, naopak to ale není vždy možné. Nemůžete vždy použít editory LAD nebo FBD pro zobrazení programu, který byl napsán v editoru STL.

2.2 Operátorský panel SIMATIC TP 177 micro

Tato kapitola vychází z [5]

2.2.1 Základní popis operátorského panelu SIMATIC TP 177 micro

Tento operátorský panel je nejvýkonnějším modelem micropanelů od firmy Siemens. Jeho dotykový panel o velikosti 5,7“ a rozlišení 320 x 240 obrazových bodů, nově umožňuje využívat vektorovou grafiku ve čtyřech odstínech modré. Lze jej používat jak horizontálně tak i vertikálně, tedy na šířku i na výšku. Pro komunikaci je panel vybaven portem RS 485, který zároveň podporuje sběrnice RS 232 a RS422. Napájen je stejně jako automat 24V. Programové vybavení do tohoto panelu se vytváří pomocí WinCC flexible micro.



Obr. 2.6 Operátorský panel SIMATIC TP 177 micro [4]

2.2.2 Technické parametry operátorského panelu SIMATIC TP 177 micro

Napájecí napětí	24V DC(+20,4V – 28,8V)
Jmenovitý proud	0,24A
Flash paměť pro uživatelská data	256KB
Připojení	RS 485 (max. 187,5Mbit/s)
Podporované sběrnice	RS232, RS422, RS485
Krytí	IP65/IP20

Tabulka 2 Technické parametry operátorského panelu [5]

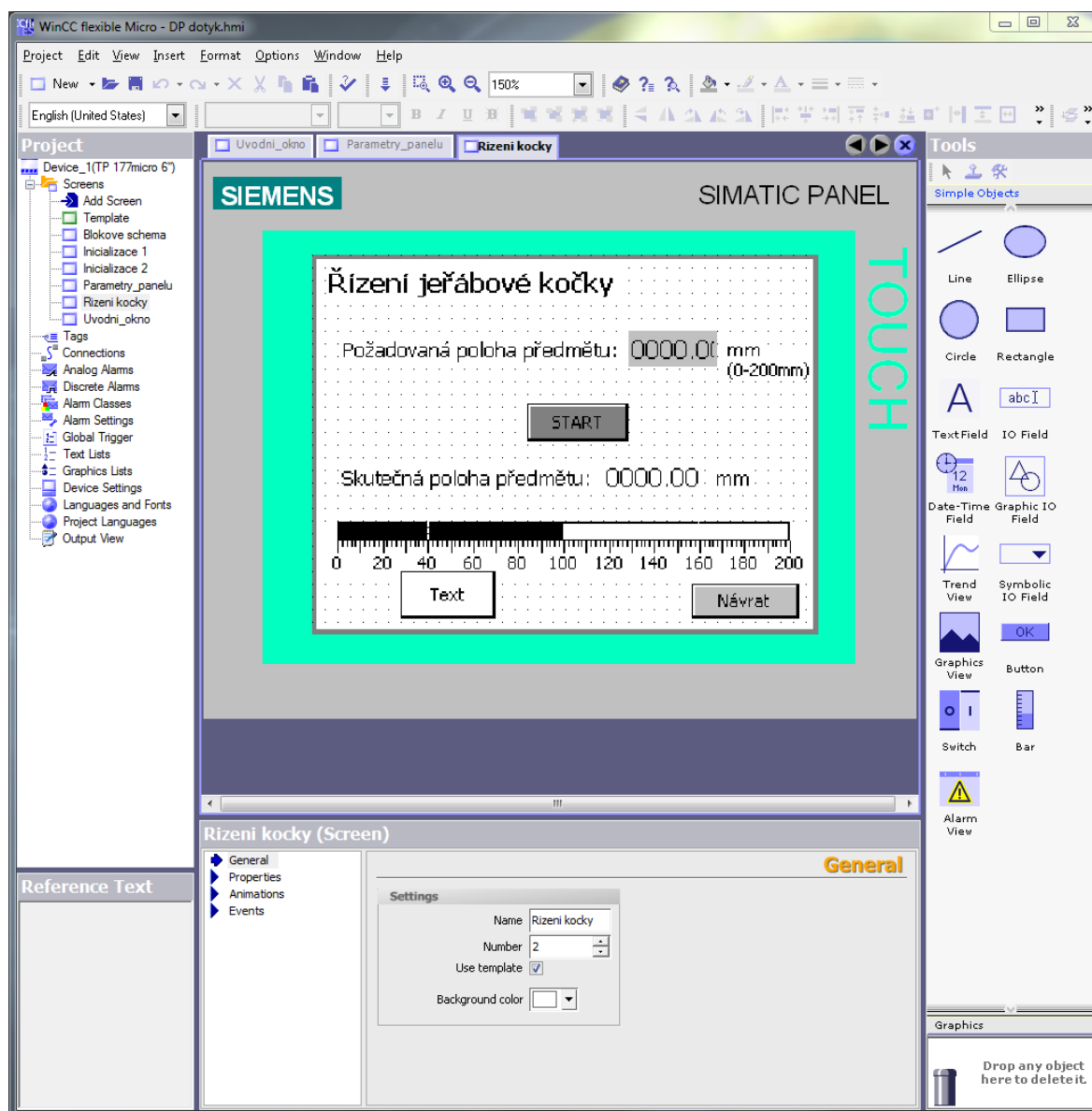
2.2.3 Vytvoření programu pomocí SIMATIC WinCC flexible 2008 Micro

Vývojové prostředí WinCC flexible 2008 Micro (Obr. 2.7) patří mezi nejnižší řadu WinCC flexible. Je schopný vytvářet a upravovat projekty pro operátorská rozhraní HMI (Human-Machine-Interface) pro panely řady Micro. Projekty vytvořené pomocí „micro“ lze otevřít a editovat ve vyšších vydání WinCC flexible. Využívá se všude, kde je potřeba řídit a sledovat technologické procesy.

Spustíme-li WinCC flexible Micro, máme možnost na výběr mezi novým projektem, nebo již vytvořeným a uloženým projektem. Zvolíme-li nový projekt, musíme vybrat, pro jaký panel budeme projekt vytvářet, a zdali bude orientovaný na výšku nebo na šířku. V mém případě jsem zvolil panel TP177micro 6“ orientovaný horizontálně. Hned poté se otevře vývojové prostředí. Nahoře, stejně jako u Step 7-MicroWIN a jiných programů, je lišta s menu a panel s nástroji. Vlevo je okno s přehledem celého projektu. Jsou zde všechny vytvořené obrazovky panelu, proměnné (Tagy), nastavení komunikace s PLC a další. Uprostřed se nachází pracovní plocha, tedy jednotlivé obrazovky, které vytváříme. Z panelu nástrojů na pravé straně můžeme vkládat objekty na pracovní plochu, například

text, grafy, tlačítka, přepínače, ale také vlastní vytvořené objekty, grafické zobrazení různých strojních zařízení. Každý objekt má své vlastní nastavení (umístění dole uprostřed). Lze zde nastavovat, písmo, barvu, nebo funkci, kterou má daný objekt provést například na kliknutí. Objekty mohou být statické (rámečky, čáry, texty), a dynamické. Ty se mohou měnit, například při stisknutí, nebo se mění v závislosti na běhu programu.

Proměnné neboli Tagy jsou interní nebo externí. Interní fungují uvnitř programu a odkazují například na jednotlivé obrazovky. Externí jsou pro komunikaci s programovatelným automatem a mají stejné datové typy jako PLC.



Obr. 2.7 Vývojové prostředí WinCC flexible Micro

2.3 Frekvenční měnič SINAMICS G110

Tato kapitola vychází z [7]

2.3.1 Základní popis frekvenčního měniče SINAMICS G110

SINAMICS G110 patří mezi základní řadu frekvenčních měničů firmy Siemens. Je vhodný pro malé a střední průmyslové aplikace, kde je zapotřebí řídit rychlost. Jako například řízení ventilátorů, čerpadel, automatizovaných balících linek, pásových dopravníků a mnoho jiných. Lze jej také použít pro domácí použití, například ovládání vrat, brán a v podobných domácích aplikacích.

Tento měnič umožňuje lineární napěťově frekvenční řízení (U/f), kvadratické U/f řízení, multibodové řízení U/f. Je napájen jednofázovým střídavým napětím 1 x 230V o frekvenci 50 ale i 60Hz. Výstup tohoto měniče je třífázové napětí 3 x 230V a je použitelný pro výkony od 0,12kW do 3kW. Dobře odolává elektromagnetickému rušení, snadno se instaluje, pro snížení hluku používá měnič vysokou frekvenční modulaci. Má možnost nastavení doby rozběhu a doběhu. SINAMICS G110 lze pořídit v několika verzích. Verze řízená analogově (napětím 0-10V) nebo pomocí ovládacího panelu BOP a verze s univerzálním rozhraním USS kde se pro komunikaci využívá protokol RS485. Dále verze s filtry či bez filtrů. Mnou použitá varianta je použitelná pro maximální výkon 0,25kW a využívá rozhraní USS.



Obr. 2.8 Frekvenční měnič SINAMICS G110 [6]

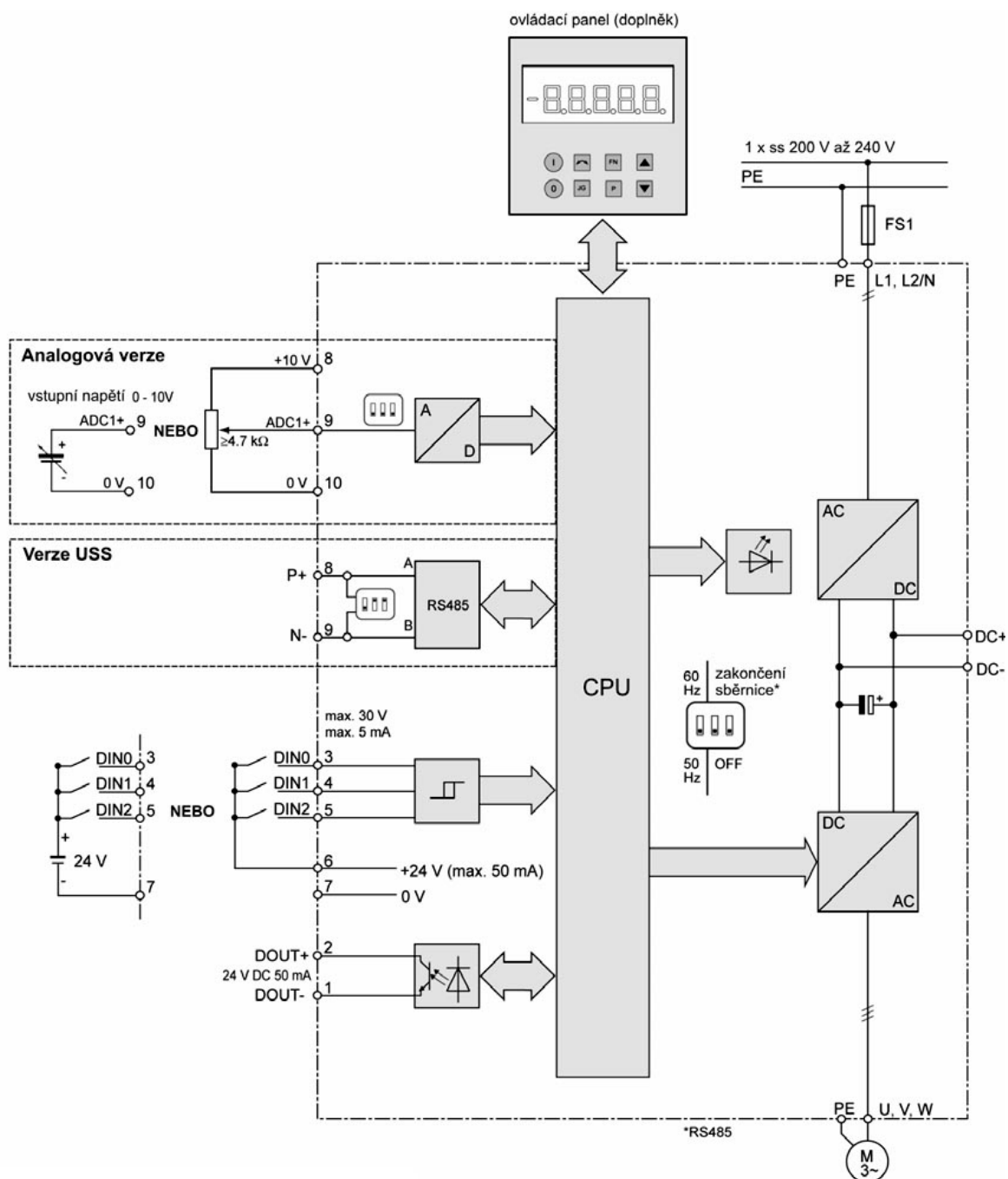
2.3.2 Technické parametry měniče SINAMICS G110

Vstupní napětí	200÷230V ± 10% 1AC
Rozsah výkonů	0,12 ÷ 3kW
Vstupní proud	4,5A
Výstupní proud	1,7A
Vstupní frekvence	47 ÷ 63Hz
Výstupní frekvence	0 ÷ 650Hz
Účinnost měniče	U výkonů < 750W je účinnost 90 ÷ 94 % U výkonů ≥ 750W je účinnost ≥ 95 %
Modulační frekvence	8 kHz (standardně); 2÷16 kHz (s krokem 2kHz)
Ochrany	Podpětí, přepětí, zkrat na zem, zkrat mezi fázemi, chod bez zátěže, ochrana motoru I _{2t} , přehřátí motoru i měniče
Krytí	IP20

Tabulka 3 Technické parametry měniče Sinamics G110 [7]

2.3.3 Blokové schéma měniče a jeho popis

Schéma na Obr. 2.9 znázorňuje vnitřní zapojení frekvenčního měniče. Nahoře je znázorněn BOP panel, který je dle potřeby odnímatelný. Tímto panelem je možné zapínat a vypínat motor, měnit frekvenci, rotaci ale i parametry měniče. Vlevo je znázorněna řídící svorkovnice. U analogové verze je možné zapojit na svorky 8, 9 a 10 potenciometr nebo řídící stejnosměrné napětí v rozmezí 0 – 10V. U verze USS je pro komunikaci zapojena sběrnice RS485 na svorky 9 a 10, tedy kanál A a kanál B. Zbylé svorky 1 – 7 můžeme použít pro další ovládání měniče. Svorka 1 a 2 slouží jako digitální výstup. Svorky 3, 4 a 5 sklouží jako digitální vstupy. Standardně je svorka 3 nastavena pro rychlé zastavení motoru, svorka 4 pro reverzi motoru a svorka 5 pro potvrzení poruchy. Lze je podle manuálu také nastavit na jiné funkce. V Tabulka 4 je souhrnný popis jednotlivých svorek frekvenčního měniče.



Obr. 2.9 Schéma vnitřního zapojení frekvenčního měniče [7]

Svorka	Popis	Funkce
1	DOUT –	Digitální výstup (-)
2	DOUT +	Digitální výstup (+)
3	DIN0	Digitální vstup 0
4	DIN1	Digitální vstup 1
5	DIN2	Digitální vstup 2
6	24V	Izolovaný výstup +24V/50mA
7	0V	Výstup 0V
Varianta	USS	Analogová
8	RS485 P+ (A)	Výstup +10V
9	RS485 N- (B)	Analogový vstup (ADC)
10	Výstup 0V	

Tabulka 4 Popis jednotlivých svorek frekvenčního měniče [7]

2.3.4 Popis knihovny USS

Tato kapitola vychází z [1]

Knihovny instrukcí STEP 7-Micro/WIN usnadňují řízení pohonů MicroMaster, protože obsahují již nakonfigurované podprogramy přerušení a jiné podprogramy, které jsou speciálně navrženy pro použití USS protokolu při komunikaci s pohonem. Pomocí instrukcí USS můžeme řídit fyzický pohon a číst, popřípadě zapisovat jeho parametry.

Z dřívějších aplikací vím, že škola vlastní knihovnu USS4, a také, že ji mohu použít pro komunikaci s frekvenčním měničem. Tato knihovna obsahuje 14 podprogramů, 8 instrukcí a 3 podprogramy přerušení. Mezi instrukce, které jsem nejčastěji používal, patří například instrukce USS4_INIT, ta se používá pro povolení a inicializaci, nebo pro ukončení komunikace s pohony. Dále pak instrukce USS4_DRV_CTRL pro řízení pohonu, a instrukce USS4_RPM_x/USS4_WPM_x pro čtení/zápis parametrů z pohonu. V jednu dobu může být aktivní pouze jedna instrukce pro čtení nebo pro zápis. Chceme-li do pohonu odeslat instrukci, například pro řízení, tedy USS4_DRV_CTRL, musí obsahovat všechny potřebné parametry. Například takto:

Ukázka instrukce USS4_DRV_CTRL

LD SM0.0

A Q0.0

CALL USS4_DRV_CTRL: SBR1, V0.1, V0.3, V3.0, I0.3, V0.2, 0, VD1000, M5.0, VB13,

VW8, VD1004, Q0.1, Q0.2, Q0.3, Q0.4

Instrukce USS4_DRV_CTRL obecně:

CALL USS4_DRV_CTRL: RUN, OFF2, OFF3, F_ACK, DIR, Drive, Type, Speed_SP, Resp_R, Error, Status, Speed, Run_EN, D_Dir, Inhibit, Fault

Popis jednotlivých instrukcí:

Bit Q0.0 signalizuje dokončení instrukce USS4_INIT, je-li tato instrukce hotová, může proběhnout instrukce USS4_DRV_CTRL.

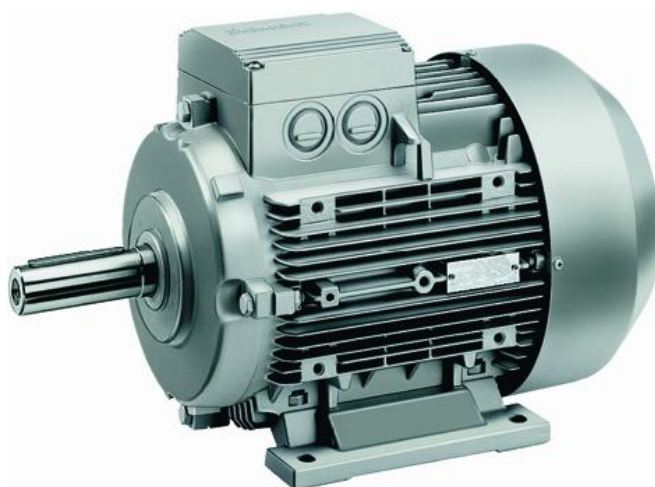
RUN	-pohon je zapnut pokud je v log. 1
OFF2	-volný doběh motoru
OFF3	-okamžité zastavení motoru
F_ACK	-vymaže poruchu přechodem z log. 0 na log.1
DIR	-změna rotace motoru
Drive	-adresa pohonu
Type	-typ motoru
Speed_SP	- požadovaná rychlost otáčení motoru
Resp_R	-potvrzení odpovědi pohonu
Error	-může obsahovat chybu pro tuto instrukci
Status	-stavové slovo odeslané zpět pohonem
Speed	-aktuální rychlost otáčení motoru
Run_EN	-indikace chodu motoru
D_Dir	-indikace směru pohonu
Inhibit	-indikace stavu blokace motoru
Fault	-indikace poruchy

2.4 Asynchronní motor s impulzním snímačem otáček

Tato kapitola vychází z [8]

2.4.1 Základní popis asynchronního motoru

Asynchronní motor je točivý elektrický stroj (elektromotor) pracující na střídavý proud. Je to nejrozšířenější pohon v elektrotechnice vůbec. Tok energie mezi hlavními částmi motoru (stator a rotor) je realizován výhradně pomocí elektromagnetické indukce, proto se často tento motor označuje jako motor indukční. Výhodou asynchronního motoru je vysoká spolehlivost, jednoduchá konstrukce a napájení z běžné střídavé sítě. Napájecí napětí může být jednofázové nebo trojfázové, přičemž trojfázové je výrazně používanější. V mém případě používám trojfázový motor od společnosti SIEMENS.

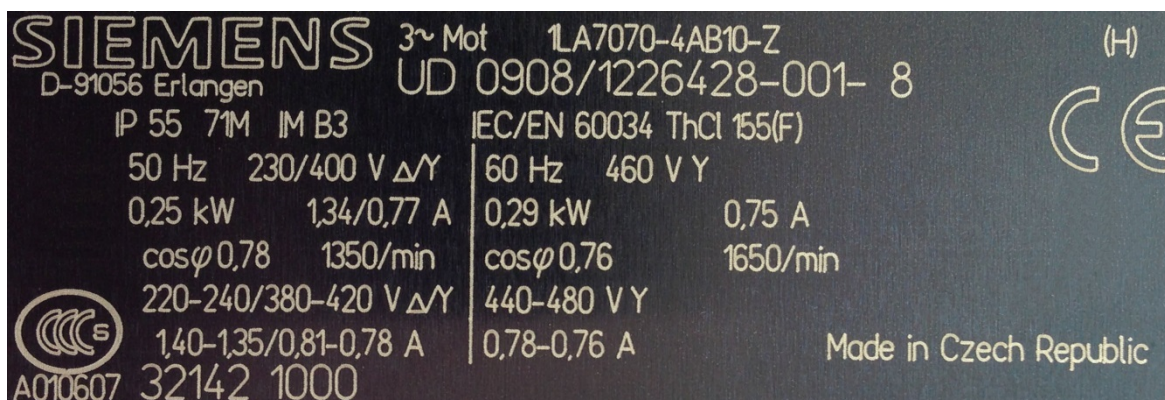


Obr. 2.10 Třífázový asynchronní motor SIEMENS [9]

2.4.2 Technické specifikace třífázového asynchronního motoru SIEMENS

Tato kapitola vychází z [10]

Firma Siemens nabízí celou řadu motorů s označením 1LA7. Liší se například osovou výškou, tvarem, počtem pólů, napětím a dalšími parametry. Tyto motory lze zapojit do hvězdy nebo trojúhelníku. Na tomto zapojení následně závisí jmenovité napětí a proud. Na štítku (viz Obr. 2.11) motoru jsou uvedeny jednotlivé elektrické a mechanické údaje. Například, že motor může být poháněn napětím 230V nebo 400V, kmitočtem 50 i 60Hz, dále pak jmenovité otáčky, napětí a podobně. Mnou použitý motor (1LA7 070-4AB10-Z) je napájen napětím 3x230V AC, proudem 1,34A, kmitočtem 50Hz a je zapojen do trojúhelníku.



Obr. 2.11 Štítek asynchronního motoru SIEMENS

Účinnost η 100%/75%	60%/60%
Moment setrvačnosti J	0,0006 kg m ²
Jmenovitý moment	1,8Nm

Tabulka 5 Technické údaje motoru Siemens [10]

2.4.3 Impulzní snímač otáček

Tato kapitola vchází z [10]

K motoru je na straně ventilátoru připevněn impulzní snímač otáček 1XP8 001-1 (zkrácené označení HTL). Je vhodný pro použití s frekvenčními měniči. Motor, který popisují výše, je již vybavený tímto snímačem. Tento snímač lze použít i na ostatní motory z této řady motorů Siemens. Jedná se o standardní snímač otáček se třemi kanály. Jeden nulový impuls Z, který ve své práci nevyužívám, a dva kanály A a B, které generují obdélníkové impulsy. Počet impulsů na otáčku je 1024 a jsou proti sobě posunuty o 90° elektrických. Díky tomu jsme schopni ze snímače odečítat jak polohu, tak i směr otáčení.



Obr. 2.12 Impulzní snímač otáček [11]

Vstupní napětí	+10V až +30V
Výstupní proud naprázdno	200mA
Výstupní proud při zatížení	max. 100mA
Počet impulsů na otáčku	1024
Výstupy	2 obdélníkové impulsy A,B 2 invertované obdélníkové impulsy A,B nulový impuls a invertovaný nulový impuls
Posun mezi impulsy 2 výstupů	$90^\circ \pm 20\%$
Maximální otáčky	9000min^{-1}
Krytí	IP66

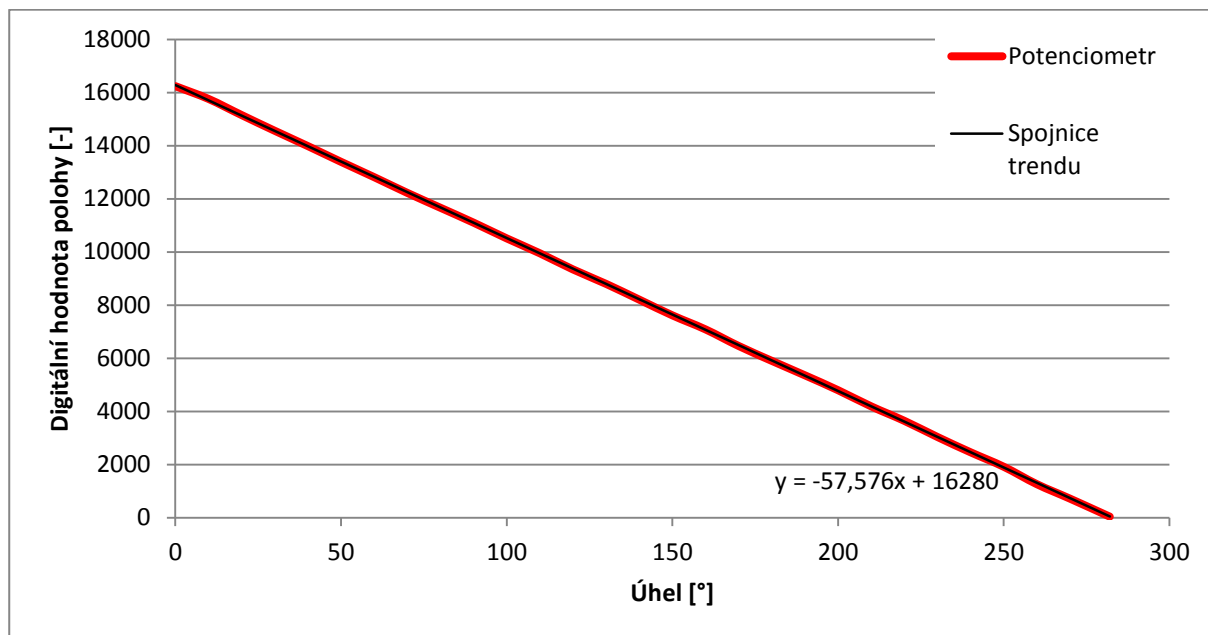
Tabulka 6 Technické parametry impulzního snímače otáček [10]

2.5 Odměrování polohy tělesa

Pro odměrování polohy zavěšeného předmětu, respektive úhlu φ (úhel vychýlení předmětu od své rovnovážné polohy), který přepočítávám na polohu tělesa, je zde použit potenciometr. Ten zároveň slouží jako osa otáčení „lana“ s tělesem, ale také jako nosné

uložení této osy. Tento potenciometr má rozsah odporu $0\Omega - 100\Omega$. Díky stupnici na potenciometru se stupnicí s rozlišením 2° jsme schopni odečítat úhel. Ten je však dorazy omezen od 0° do 272° . Protože k potenciometru jsem nesehnal žádnou dokumentaci, musel jsem si změřit, zda se jedná o potenciometr lineární-N, logaritmický-G či exponenciální-E.

Pro celou soustavu používám jeden zdroj napětí (24V DC). Maximální napětí, které je automat schopen na svém analogovém vstupu měřit, je 5V. Proto byl pro snížení napětí navržen před potenciometr odpor o velikosti 390Ω . Díky tomu je na svorkách napětí 5V, respektive na jezdcí potenciometru jsem schopen měřit napětí 0-5V. Toto napětí je schopen automat převést na digitální hodnotu o délce word, tedy 16bitů. Pro změření typu jsem tedy na potenciometru nastavoval úhel po 10° a analogové hodnoty, které automat převedl na digitální, jsem zapisoval do tabulky, podle níž jsem vytvořil graf (Obr. 2.13). Ten jsem proložil lineární spojnici trendu a dostal jsem rovnici přímky ($y = -57,576x + 16280$). Z této rovnice pak dále používám směrnici přímky pro přepočítání hodnot z AD převodníku na úhel ve stupních tak, aby odpovídal jednotlivým hodnotám na stupnici potenciometru. Z grafu je patrné, že potenciometr je lineární.



Obr. 2.13 Závislost číselných údajů na úhlu natočení potenciometru

3 Popis celé laboratorní soustavy

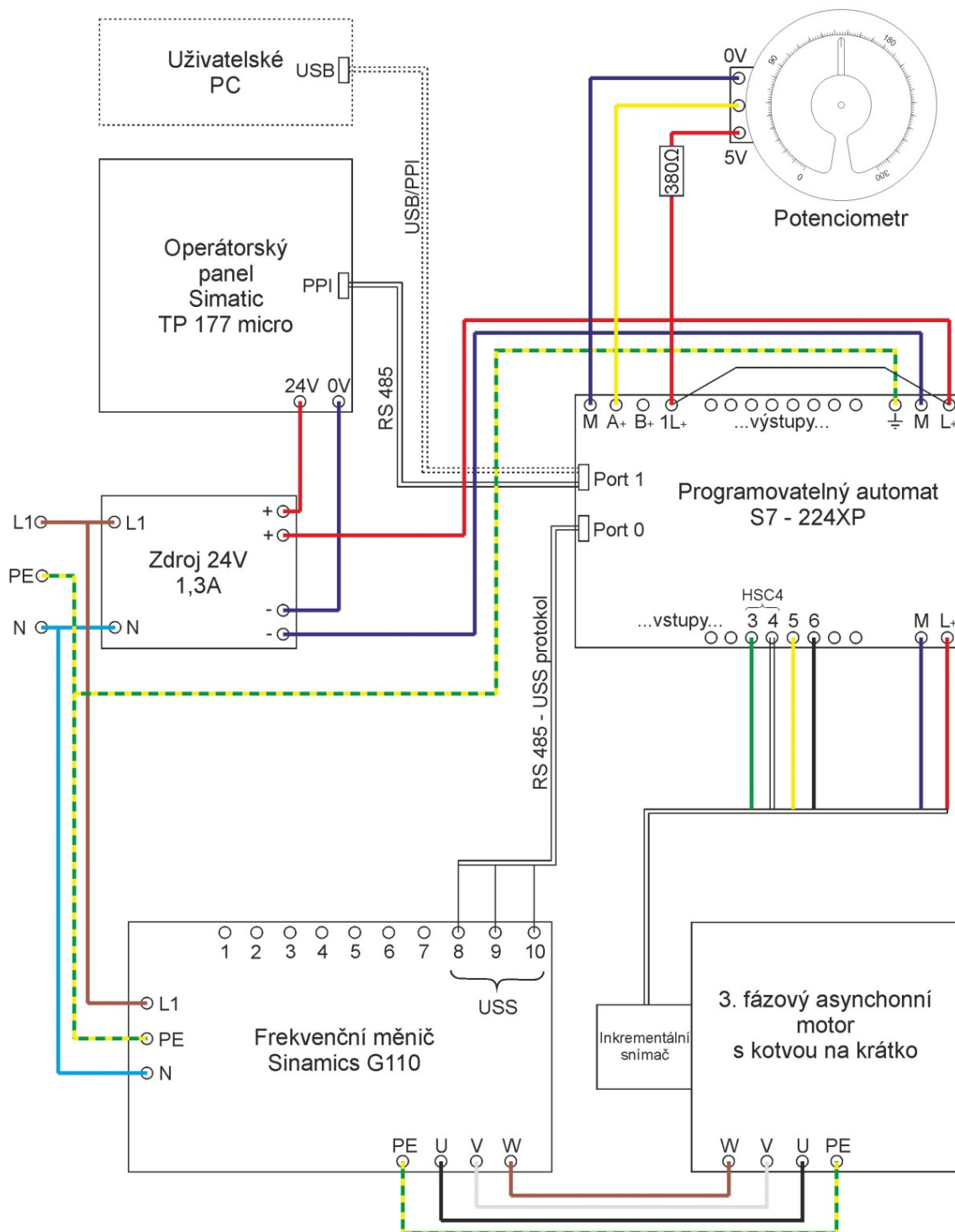
3.1 Blokové schéma laboratorní soustavy a její popis

Blokové schéma na Obr. 3.1 znázorňuje zapojení jednotlivých komponentů laboratorní soustavy. Vpravo nahoře je programovatelný automat, který komunikuje s ostatními prvky. S operátorským panelem (vlevo nahoře) komunikuje pomocí sběrnice RS 485. Ta je zakončena konektory typu CAN 9. Na straně panelu je pouze jeden konektor pro zapojení. Na straně automatu je operátorský panel zapojen na PORT 1. Díky omezenému počtu portů zároveň slouží pro komunikaci s PC (vlevo nahoře). Proto nelze při nahrávání programu do automatu a sledování programu zadávat příkazy z operátorského panelu. To stěžuje ladění programu. Pro komunikaci s PC slouží speciální kabel USB/PPI. Pomocí něj můžeme nahrávat program nejen do automatu, ale také do operátorského panelu.

Druhý port, tedy PORT 0 slouží protokolu USS, a tedy komunikaci s frekvenčním měničem. Jedná se také o sběrnici RS 485, která je na straně automatu zakončena konektorem CAN 9 a na druhé je zapojena na svorky frekvenčního měniče (viz kapitola 2.3.3). Frekvenční měnič je napájen 230V~. Ten pak dále napájí a řídí 3-fázový motor s kotvou nakrátko (vpravo dole) třífázovým napětím s proměnným kmitočtem a napětím.

Zpětná vazba, tedy informace o poloze a rychlosti, je hodnota kterou předává automatu inkrementální snímač. Ten je připojen k automatu na svorky digitálních vstupů. Svorky 3 a 4 jsou svorky vysokorychlostního čítače HSC4, k nimž jsou připojeny kanály A a B. Nulový kanál a diagnostický kanál je připojen na vstupy 5 a 6, ty však nevyužívám. Inkrementální snímač je napájen 24V zdrojem (vlevo uprostřed), stejně jako automat a operátorský panel.

Napětí jezdce potenciometru (0-5V), jakožto informace o poloze tělesa, je připojeno k automatu na analogový vstup na svorku A+. Potenciometr, jak jsem již zmínil v kapitole 2.5, je napájen 24V redukovánými odporem na 5V.



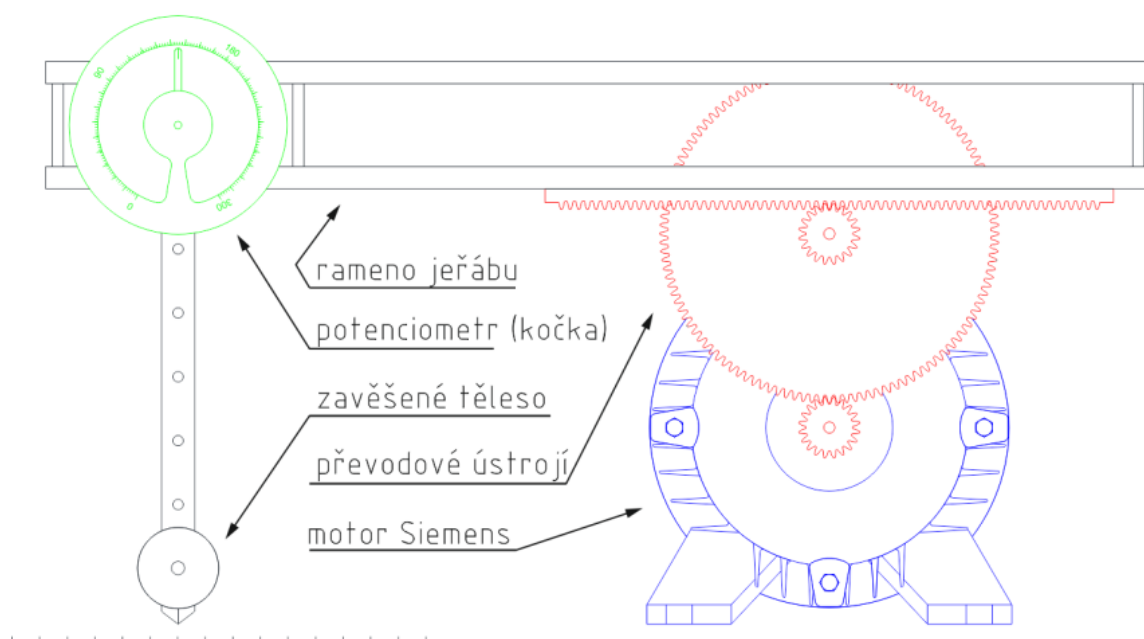
Obr. 3.1 Blokové schéma laboratorní soustavy

3.2 Schéma mechanismu kočky a její popis

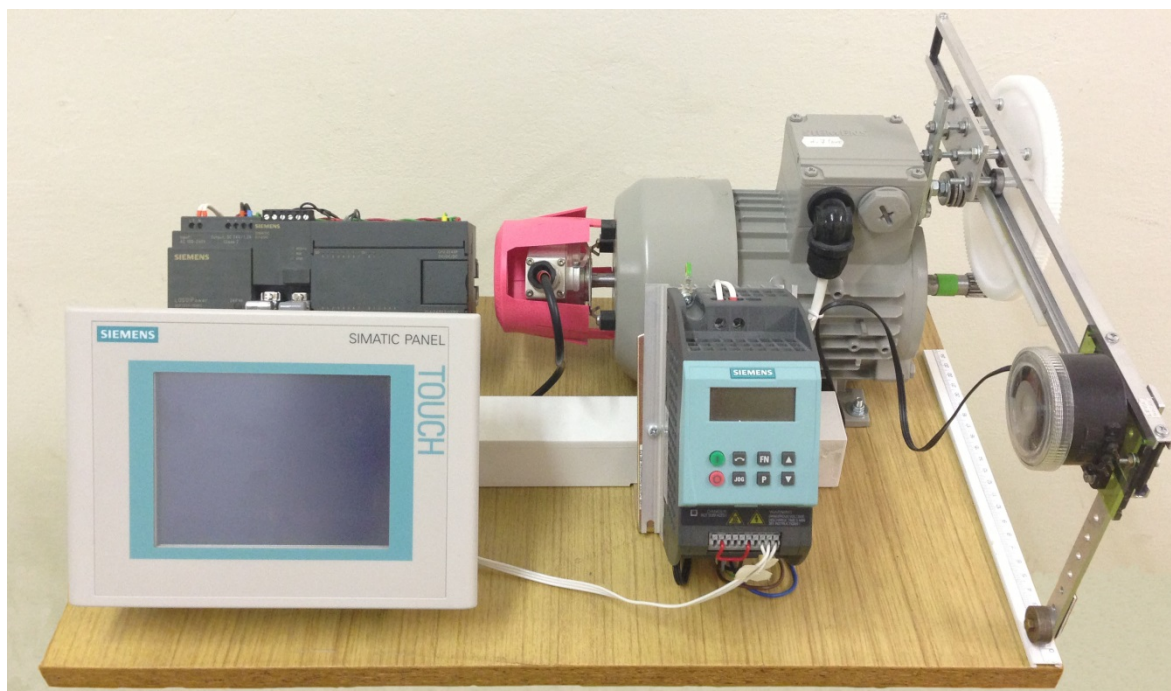
Na Obr. 3.2 je znázorněno schéma celého mechanismu kočky. Tento mechanismus má obrácenou konstrukci kočky, než bývá zvykem. V praxi totiž kočka, tedy motor s navijákem, jezdí na rameni jeřábu. Tato konstrukce je však příliš velká a proto pro laboratorní podmínky byl vytvořen model, kde se pohybuje rameno jeřábu a pohon stojí na místě. Jako kočku si tedy můžeme představit potenciometr se zavěšeným tělesem.

Na výstupní hřídeli motoru je přišroubován ozubený pastorek s 15-ti zuby. K tomuto pastorku přiléhá ozubené kolo se 140-ti zuby. Převodový poměr je tedy 15: 140. Rozteč zubů je 3mm. Toto kolo stejně jako vedení ramene je připevněno ke konzoli přišroubované k motoru. Součástí velkého ozubeného kola je malé ozubené kolo mající také 15 zubů. To přiléhá k ozubenému hřebeni na rameni „jeřábu“. Početně i experimentálně jsem tedy naměřil, že jedním otočením pastorku o 360° se rameno posune o 5mm.

Na konci ramene je umístěn potenciometr. Osu otáčení tohoto potenciometru beru jako vzdálenost kočky od referenční polohy, dále jen poloha kočky. Tento typ potenciometru byl zvolen proto, že má oproti ostatním velmi nízké tření, a zavěšené těleso se tedy může houpat s minimálním odporem. Toto těleso je od osy otáčení vzdáleno 176mm. Polohu tělesa si orientačně mohou odečítat z pravítka umístěného pod ním.



Obr. 3.2 Schéma mechanismu kočky a závěsu



Obr. 3.3 Ukázka celého laboratorního setu

4 Návrh regulace polohy zavěšeného tělesa

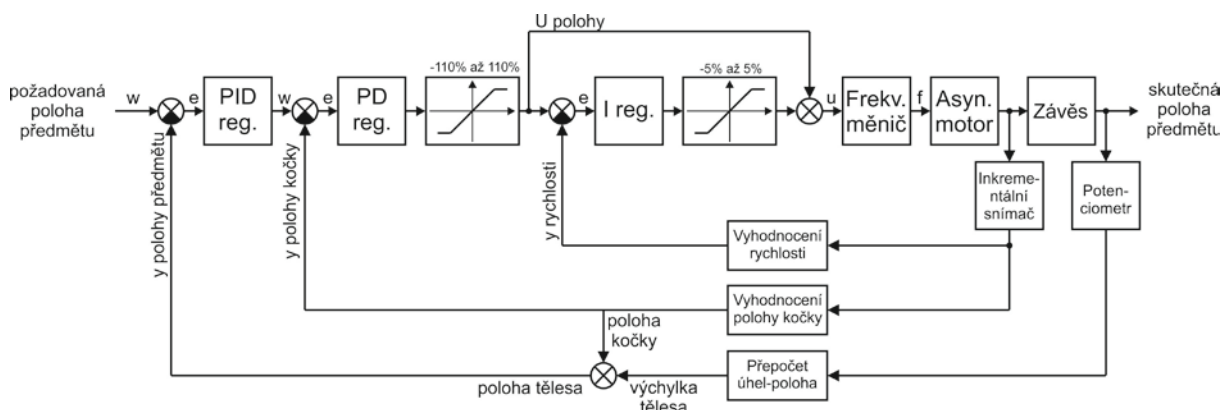
Základní myšlenkou bylo vytvoření rozvětveného regulačního obvodu, který by byl schopen přemísťovat zavěšené těleso takovým způsobem, aby docházelo k co nejmenšímu kmitání. Rozvětvená regulace je volena proto, že se jedná o vysoce kmitavou soustavu a je zapotřebí co nejkvalitnější regulace. Z práce [3] využiji znalosti pro řízení polohy motoru a využiji některá dílčí řešení. Navrhnou rozvětvený regulační obvod, ke kterému použiji další varianty regulace polohy tělesa.

4.1 První varianta řízení

První varianta navrženého regulátoru (viz. Obr. 4.1) spočívala v předřazení PSD regulátoru polohy tělesa před regulátor polohy PD s vnitřní rychlostní regulační smyčkou. Pro regulaci rychlosti jsem použil S regulátor, který spíše slouží pro kompenzaci skluzu motoru, proto je akční zásah omezen v rozmezí -5% až 5%. K tomuto akčnímu zásahu je přičten akční zásah polohového PD regulátoru. Tato takzvané dopředné řízení zvyšuje rychlost celé regulace. Takto upravený akční zásah v podobě rychlosti je posílán do frekvenčního měniče pomocí USS protokolu.

Odchylka pro rychlostní regulátor je rozdílem akčního zásahu polohového regulátoru a zpětné vazby rychlosti získávané z inkrementálního snímače. Akční zásah polohového regulátoru je omezen v rozmezí -110% až 110%. Toto omezení je zde proto, že akční zásah může nabývat i velmi vysokých hodnot a frekvenční měnič by je nemusel akceptovat. 100% odpovídá hodnota 50Hz. Maximální frekvence je tedy 55Hz.

Požadovaná poloha tělesa je zadávána v milimetrech uživatelem pomocí operátorského panelu. Skutečná poloha tělesa vznikne přepočtem úhlu pomocí funkce sinus na vzdálenost, o jakou je těleso vychýleno a k této hodnotě přičtu polohu „kočky“. Rozdílem požadované a skutečné polohy tělesa dostaneme odchylku pro PSD regulátor polohy tělesa. Akční zásah tohoto regulátoru vstupuje do rozdílového členu před PD regulátorem polohy kde se tento akční zásah odečítá od skutečné polohy kočky. Z něj vstupuje do tohoto regulátoru odchylka polohy.



Obr. 4.1 Schéma první varianty řízení

Celá regulace probíhá nespojitě v čase, tedy s číslicovými hodnotami. Je to dáno tím, že regulátor je realizovaný softwarově v PLC. Číslicové hodnoty nemůžeme derivovat ani

integrovat jako u spojitých regulátorů. Proto je integrace zaměněna za sumaci a derivace je nahrazena diferencí.

Proto akční zásah S regulátoru rychlosti je dán vztahem:

$$u(k) = u(k-1) + \left[K_p \frac{T}{T_i} \right] e(k) \quad (4.1)$$

kde $u(k)$ je akční zásah S regulátoru, $u(k-1)$ je hodnota předchozího akčního zásahu, $e(k)$ je odchylka mezi skutečným a požadovaným stavem, K_p je proporcionální zesílení, často také udávané jako r_0 . V tomto případě je rovno 1, protože se jedná pouze o S regulátor bez proporcionální složky. Dále pak T je vzorkovací perioda a T_i je hodnota, kterou určujeme pomocí metod pro seřízení číslicových regulátorů, například metodou Ziegler-Nichols.

Akční zásah PD regulátoru polohy je definován vztahem:

$$u(k) = K_p e(k) + \left[K_p \frac{T_d}{T} \right] [e(k) - e(k-1)] \quad (4.2)$$

kde $u(k)$ je akční zásah PD regulátoru, K_p je proporcionální zesílení, T_d je derivační časová konstanta. Obě tyto hodnoty jsou určeny metodami pro seřízení číslicových regulátorů. Dále pak T je vzorkovací perioda, $e(k)$ je regulační odchylka a $e(k-1)$ je předešlá regulační odchylka.

Vztah pro výpočet akčního zásahu PSD regulátoru polohy tělesa:

$$u(k) = K_p e(k) + \left\{ u(k-1) + \left[K_p \frac{T}{T_i} \right] e(k) \right\} + \left[K_p \frac{T_d}{T} \right] [e(k) - e(k-1)] \quad (4.3)$$

kde jednotlivé proměnné jsou stejné jako v předešlých rovnicích.

4.1.1 Nastavení regulátoru

Pro dobrý chod celé soustavy bylo zapotřebí seřadit jednotlivé regulátory. Pro seřízení regulátorů jsem zvolil metodu Ziegler – Nichols. Princip této metody je takový, že vyřadíme integrační a derivační složku regulátoru a postupným zvyšováním zesílení r_0 přivedeme regulovaný obvod na hranici stability. Takovéto zesílení se nazývá kritické zesílení r_{0k} , označované též jako K_{PK} . To je stav, kdy soustava kmitá netlumenými kmity o stejné amplitudě. Změřením periody těchto kmitů získáme kritickou periodu T_k .

Vždy, když nastavujeme rozvětvený regulační obvod touto metodou, začínáme nejprve u vnitřní smyčky a všechny ostatní regulační smyčky vyřadíme. V tomto případě jsem ponechal pouze S regulátor rychlosti. Tento obvod bychom měli přivést na hranici stability změnou integrační konstanty T_i , takto získaná kritická hodnota se poté nazývá T_{ik} . Tento obvod však nelze přivést na hranici stability, muselo by se jednat o soustavu třetího řádu. V takovém případě se vkládá do programu setrvačný člen, který při odstranění příliš nezmění regulovanou soustavu. V práci [3] byla tato hodnota již stanovena, proto jsem ji použil a zkorigoval. Měřením jsem stanovil tuto integrační konstantu na $T_i=0,082$ s. Doba vzorkování rychlostního regulátoru je $T=0,010$ s.

K takto nastavené regulační smyčce jsem připojil druhý regulátor v pořadí, tedy PD regulátor polohy. Vyřazením derivace ponecháme pouze proporcionální konstantu, kterou stejně jako u předešlého regulátoru přivedeme na hranici stability. Hodnota kritického zesílení byla $r_{0k}=66$ a kritickou periodu jsem změřil na hodnotu $T_k=0,56$ s. Výpočtem z

[13] jsem poté stanovil zesílení na hodnotu $r_0=26,4$ (K_p) a derivační konstantu $T_d = 0.028$ s. Doba vzorkování polohového regulátoru je $T=0,015$ s.

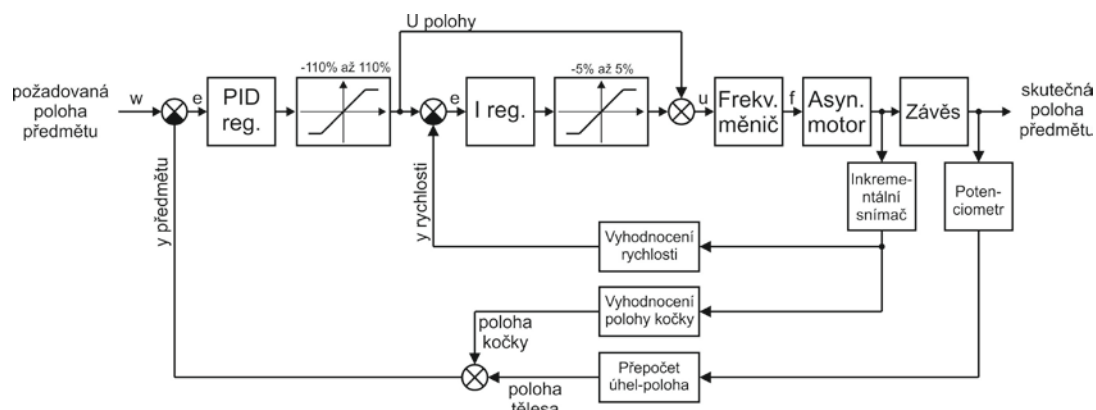
Takto nastavené regulační smyčky rychlosti a polohy vykazovaly poměrně dobré vlastnosti při regulaci polohy kočky. Proto jsem připojil třetí regulátor, tedy PSD, pro regulaci polohy tělesa. Ten je v programu umístěn v časovém přerušení spolu s řízením polohy kočky. Je to proto, že automat Simatic S7-224XP má pouze dvě časová přerušení. Vnitřní smyčky se většinou realizují s větší vzorkovací periodou, a proto se výpočet tohoto regulátoru polohy tělesa provádí pouze každé druhé spuštění tohoto přerušení. Tím dostanu vzorovací periodu pro regulátor polohy tělesa dvojnásobnou, tedy 30ms. Vyřadil jsem integrační a derivační složku a pokoušel se nastavit proporcionální zesílení. Soustava však byla silně nestabilní s hodnotami vyššími než 1 a pokud byla hodnota menší než jedna, těleso nedosahovalo požadovaných hodnot, a to ani přibližně. Domníval jsem se, že dochází k nesprávné interpretaci zpětných polohových vazeb kočky a tělesa. Jestliže se totiž poloha předmětu blíží k požadované, akční zásah PID regulátoru polohy tělesa se snižoval, ale na rozdílovém členu PD regulátoru polohy kočky se zpětná vazba polohy kočky zvyšovala. Díky tomu vznikne vyšší regulační odchylka i přesto, že poloha tělesa se blíží k požadované. Proto jsem od zpětné vazby skutečné polohy kočky odečítal polohu požadovanou. To by mělo mít za důsledek, že regulační odchylka by při započetí regulace byla dvojnásobná, ale postupem regulace by se snižovala. Jednalo by se také o dopředné řízení. Ale bohužel ani tato varianta neeliminovala kmitání, a soustava se chovala velice podobně. Postupným zkoušením a úpravami jsem se dostal ke druhé variantě řízení.

4.2 Druhá varianta řízení

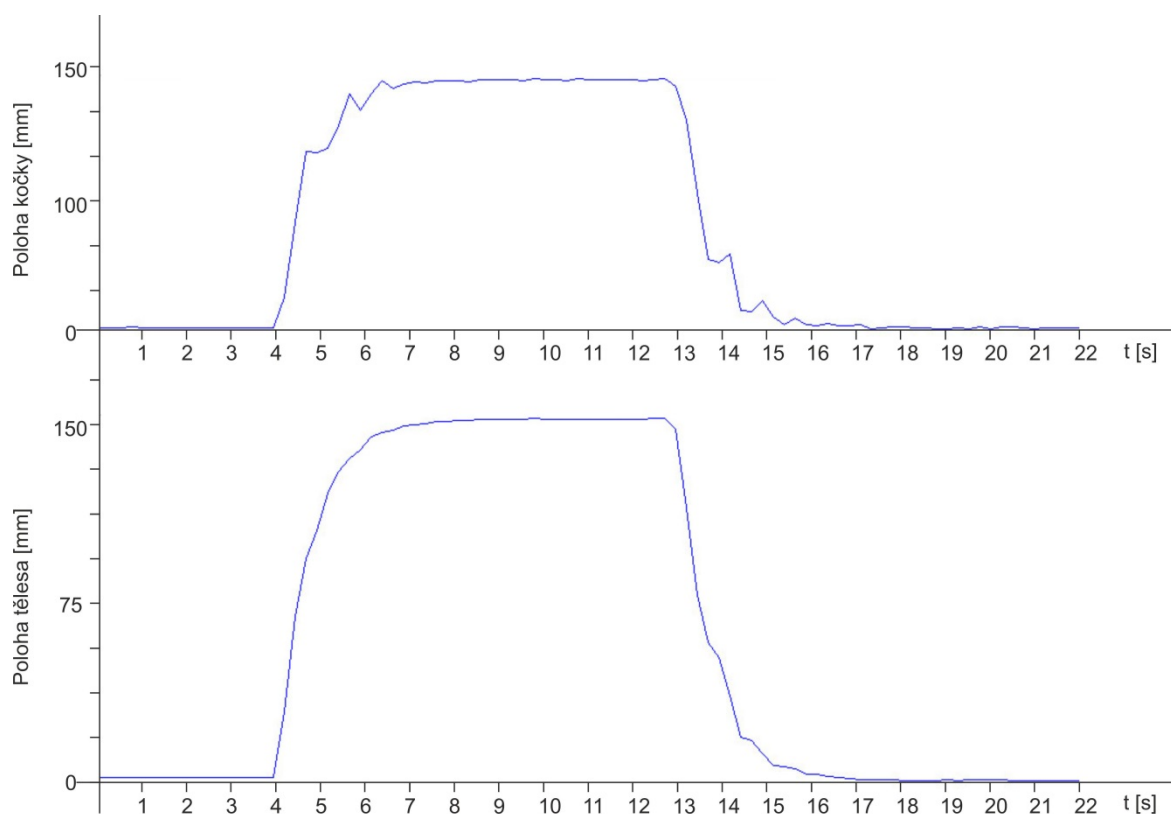
Tato varianta měla nejprve podobu první varianty, ale s tím rozdílem, že jsem dobu vzorkování PSD regulace polohy tělesa snížil na 15 ms stejně jako u regulace polohy kočky. Domníval jsem se, že by to mohlo zmenšit nestabilitu polohy. Toto opatření však nemělo téměř žádný vliv a proto jsem se rozhodl odstranit smyčku regulace kočky. Tak vznikl finální regulátor polohy tělesa znázorněný na Obr. 4.2.

Do PSD regulátoru polohy tělesa vstupuje regulační odchylka, tedy rozdíl mezi skutečnou a požadovanou polohou tělesa. Akční zásah tohoto regulátoru je taktéž omezen shora i zdola na 110%, protože při velkých regulačních odchylkách polohy by mohlo docházet k velkým akčním zásahům. Takto upravený akční zásah vstupuje do rozdílového členu rychlostního regulátoru, kde se odečte od skutečné rychlosti a vznikne regulační odchylka. Ta vstupuje do regulátoru rychlosti. Akční zásah regulátoru rychlosti, tedy rychlost otáčení, zapisuje pomocí instrukce USS do frekvenčního měniče.

Postup ladění regulátorů je podobný jako u předešlé varianty. Integrační časovou konstantu jsem ponechal stejnou, tedy $T_i=0,082$ s. Připojil jsem druhou regulační smyčku, kde jsem vyřadil integraci a derivaci, a přivedl ji na hranici stability. Kritické zesílení jsem stanovil na $r_{0k}=0.2$ a kritickou periodu na $T_k=0,83$ s. Z těchto hodnot jsem vypočítal jednotlivé složky. Proportionální zesílení $r_0=0,12$, integrační časovou konstantu $T_i=0.42$ s a derivační časovou konstantu $T_d=0.1$ s. Takto seřízený regulátor vykazoval mnohem lepší vlastnosti, než první varianta. Tedy kmitavost tělesa při jeho přemísťování je poměrně malá a nedosahuje žádného překmitu. To je patrné na Obr. 4.3. I přesto, že je regulátor schopen regulovat tuto silně kmitavou soustavu, je akční zásah příliš opožděn. To je patrné na Obr. 4.6. V kapitole 4.2.1 bude tento problém vysvětlen.



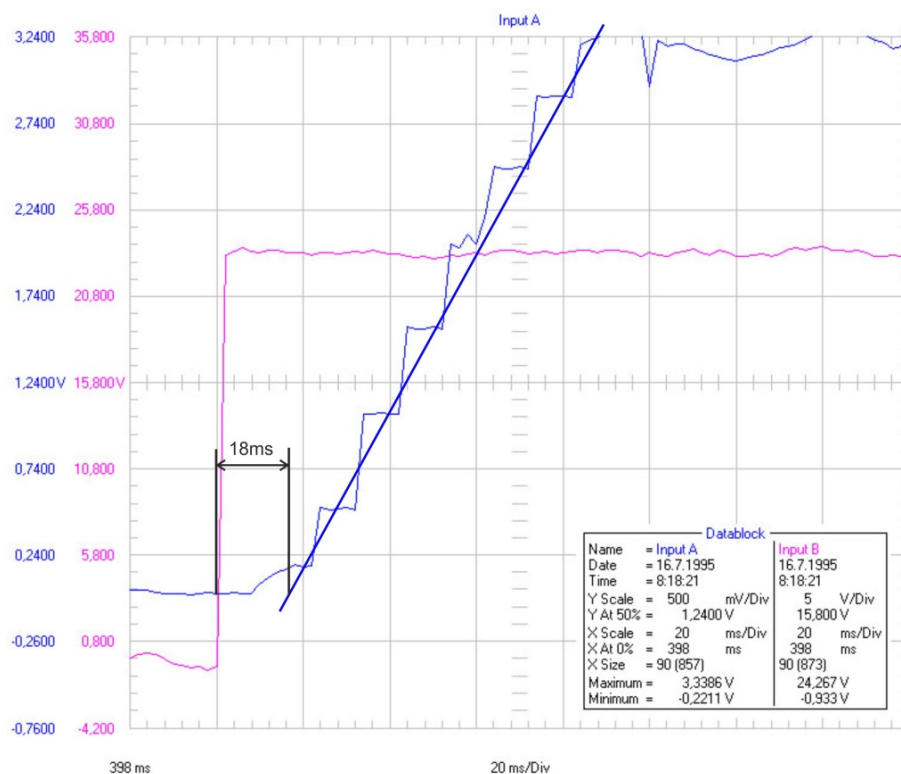
Obr. 4.2 Schéma druhé varianty řízení (použito)



Obr. 4.3 Časový průběh otáček motoru a polohy tělesa

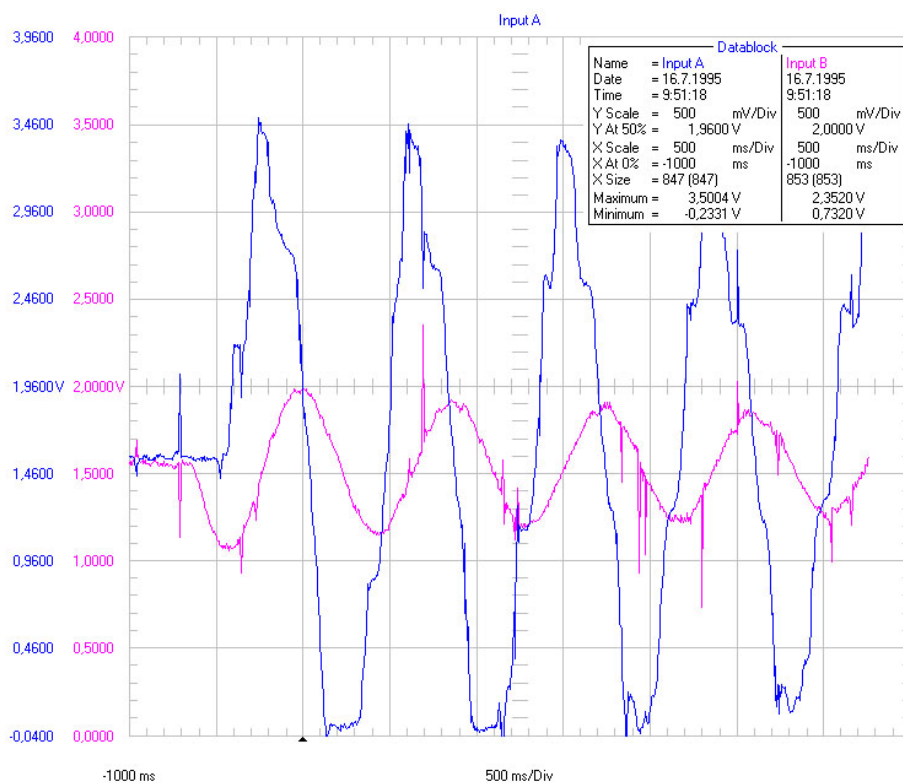
4.2.1 Měření a experimenty

Experimenty vykazovaly zdánlivě nevysvětlitelné chování, proto bylo nutné vlastnosti ověřit měřením. Nejprve jsem změřil dopravní zpoždění komunikace mezi automatem a frekvenčním měničem, které činí 18ms, viz Obr. 4.4. V tomto zpoždění je započten i čas převodu skutečných otáček na analogovou hodnotu, tento čas je 92 μ [1]. Toto měření jsem provedl tak, že jsem jeden kanál osciloskopu (červený) připojil na digitální výstup „I0.1“, který je v logické hodnotě 1 pokud je motor spuštěn a druhý kanál (modrý) na analogový výstup na kterém hodnota napětí 1V odpovídá 300ot/min.



Obr. 4.4 Časový průběh rychlosti motoru

Toto zpoždění, které je přibližně 200 ms, však nevysvětluje tomu, že by zpoždění mezi vychýlením tělesa a akčním zásahem bylo způsobeno dopravním zpožděním. Toto zpoždění je viditelné na Obr. 4.5 a jeho detailu na Obr. 4.6. Při měření jsem postupoval tak, že napětí jezdce potenciometru jsem připojil na kanál A (červený) a druhý kanál B (modrý) zůstal připojený na analogový výstup. Rychlým vychýlením tělesa jsem změnil regulační odchylku pro regulátor polohy tělesa a měřil jsem dobu, za jakou do soustavy zasáhne.

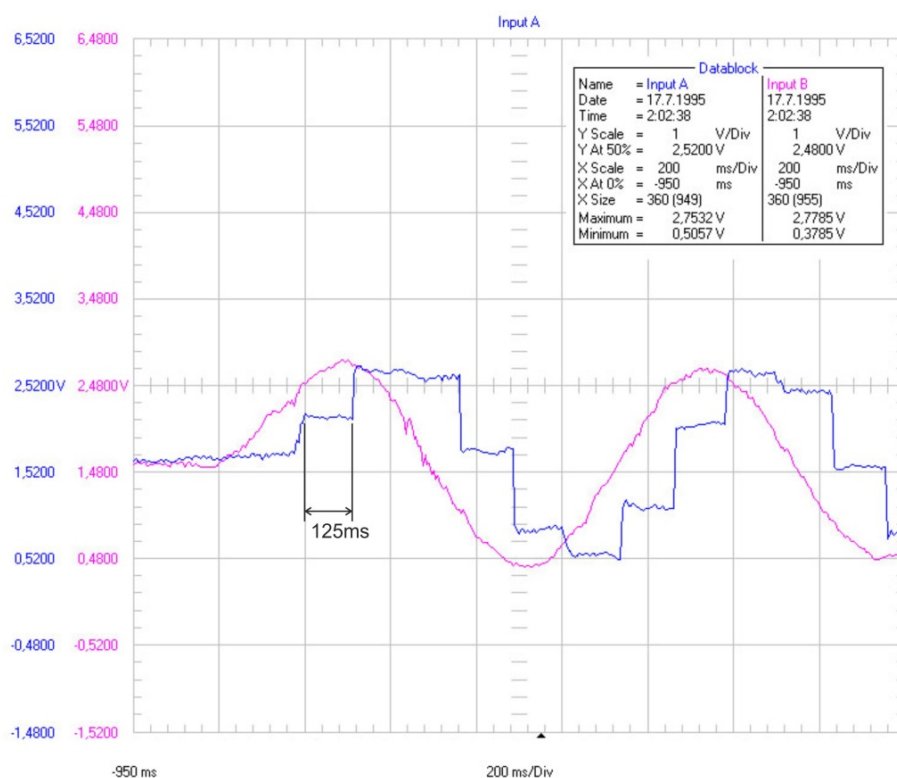


Obr. 4.5 Časové průběhy vychýlení závěsu a rychlosti motoru



Obr. 4.6 Detail časového průběhu vychýlení závěsu a rychlosti motoru

Jedinou zbylou možností je zpoždění převodu napětí jezdce potenciometru na digitální hodnotu, abych si tuto hypotézu ověřil, vytvořil jsem jednoduchý program, který analogovou hodnotu z potenciometru převede na digitální a následně ji převede na napětí analogového výstupu. Převod digitální hodnoty na analogovou trvá již zmíněných 92μ [1]. Následně jsem provedl měření (viz. Obr. 4.7), kde kanál A (červený) odpovídá hodnotám napětí z jezdce potenciometru a kanál B (modrý) odpovídá hodnotám převedeným z analogové hodnoty na digitální a zpět na analogový. Na Obr. 4.7 je patrné, že čas pro převod analogové hodnoty na digitální je 125 ms. Při podrobnějším zkoumání v [1] byly výsledky tohoto měření potvrzeny. Doba převodu analogového vstupu na digitální činí 125 ms a skoková změna může dosáhnout až 250 ms![1] Díky tomuto zjištění jsou výsledky této práce spíše pesimistické. Tento problém by mohlo vyřešit stavové řízení.



Obr. 4.7 Skutečný a převedený časový průběh napětí potenciometru

5 Pokročilé formy řízení

Tato kapitola vychází z [12]

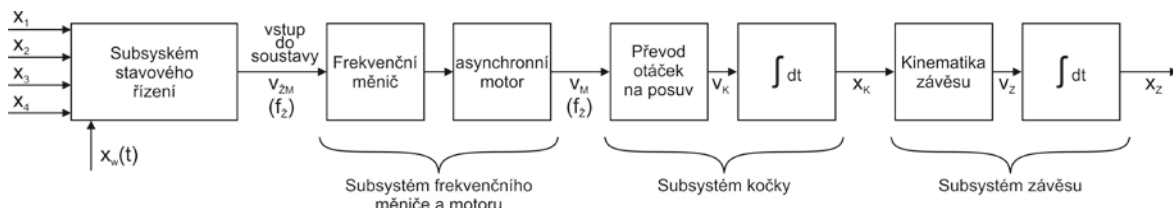
Mezi pokročilé formy řízení patří stavová a prediktivní regulace. V dnešní době jsou velmi oblíbené a rychle se vyvíjí hlavně díky tomu, že mají mnohem lepší vlastnosti než běžné regulace. Obě tyto pokročilé formy řízení jsou téměř srovnatelné, porovnáváme-li kvalitu řízení. Obě jsou však velice složité na realizaci. Obzvláště pro vytvoření programového vybavení automatu. Proto jsem tyto varianty, díky jejich náročnosti a nedostatku času již nebyl schopen realizovat. Proto alespoň popíši teoretický postup realizace.

5.1 Stavová regulace

Stavový popis zohledňuje vnitřní stav soustavy, na rozdíl od PID regulací, které pracují s popisem vycházejícím ze závislosti výstupu na vstupu. Proto jsou PID regulátory vhodné pro jednoduché systémy, například systémy s jedním vstupem a výstupem. Pro složitější systém, do kterého patří i řízení pohybu přemísťování zavěšeného tělesa, musíme brát v potaz i jiné stavové veličiny. V tomto případě rychlost kočky, rychlost zavěšeného tělesa, polohu kočky a polohu zavěšeného tělesa. Využitím těchto veličin jsme schopni určit chování systému v dalším okamžiku.

5.1.1 Stavový popis soustavy

Popis soustavy je tedy částečně odlišný od skutečného přístupu, protože zahrnuje přístrojové komponenty, které jsou již v soustavě dány. Soustavu (viz. Obr. 5.1) je vhodné rozdělit na subsystémy, které níže popíšu podrobněji.



Obr. 5.1 Blokové schéma stavové soustavy

V Obr. 5.1 vstupují do systému stavového řízení 4 stavové veličiny. Rychlost motoru $x_1 = v_m(t)$ kterou získávám z inkrementálního snímače, poloha kočky $x_2 = x_k(t)$ kterou získám integrací rychlosti motoru, rychlost zavěšeného tělesa $x_3 = v_z(t)$ kterou dostanu derivací polohy zavěšeného tělesa $x_4 = x_z(t)$. Dále pak požadovaná hodnota polohy $x_w(t)$. Výstupem je žádaná rychlost motoru v_{zm} nebo frekvence motoru f_z . Ta vstupuje do subsystému frekvenčního měniče kde výstupem je skutečná rychlost motoru v_m nebo frekvence motoru f_m . Popis této soustavy je vhodný v maticovém tvaru

$$\dot{\mathbf{x}}(t) = \underline{\mathbf{A}} \cdot \mathbf{x}(t) + \underline{\mathbf{B}} \cdot u(t) \quad (4.4)$$

$$\mathbf{y}(t) = \underline{\mathbf{C}} \cdot \mathbf{x}(t) \quad (4.5)$$

Subsystém frekvenčního měniče a motoru

Do tohoto subsystému vstupuje ze stavového řízení rychlost motoru nebo frekvence motoru. Tuto pak frekvenční měnič předá motoru. Výstupem je skutečná rychlost nebo

frekvence motoru. Reálné vlastnosti tohoto subsystému aproximujeme přenosem s dynamikou proporcionální soustavy se setrvačností 1. řádu. Tedy

$$G_{MM}(s) = \frac{v_M(s)}{v_Z(s)} = \frac{K_{MM}}{1+T_{MM} \cdot s} \quad (4.6)$$

kde, konstanta K_{MM} se vlivem skluzu motoru může měnit v rozmezí $K_{MM} \approx 0,95 \div 0,99$. Časovou konstantu pohonu s frekvenčním měničem a motorem T_{MM} získáme měřením či odhadem. V časové oblasti platí

$$v_M(t) + \dot{v}_M(t) \cdot T_{MM} = v_Z(t) \cdot K_{MM} \quad (4.7)$$

$$\dot{v}_M(t) = \frac{-1}{T_{MM}} \cdot v_M(t) + \frac{K_{MM}}{T_{MM}} \cdot v_Z(t) \quad (4.8)$$

Subsystém kočky

Vstupem tohoto subsystému je rychlost motoru, přesněji rychlost otáčení hřídele, který je převodem spojen se subsystémem kočky. Proto dynamika této soustavy zahrnuje jen integrační vztah mezi rychlostí motoru $v_m(t)$ a polohou kočky $x_k(t)$. Tedy

$$x_K(t) = K_{PŘ} \cdot \int v_M(t) dt \quad (4.9)$$

kde, $K_{PŘ}$ je konstanta převodu otáčivého převodu na posuvný. Pro účely stavového popisu převedeme na

$$\dot{x}_K(t) = K_{PŘ} \cdot v_M(t) \quad (4.10)$$

Subsystém závěsu

Tato část se chová jako matematické kyvadlo, díky tomu se jedná o silně kmitavou soustavu kterou lze popsat diferenciální rovnicí 2. řádu. Tedy

$$\ddot{x}_Z(t) = \frac{-g}{l} \cdot x_Z(t) - \frac{k_{tz}}{m_Z} \cdot \dot{x}_Z + \frac{g}{l} \cdot x_K(t) \quad (4.11)$$

kde, $x_Z(t)$ je stavová veličina absolutní polohy závěsu, l je délka lana, m_Z je hmotnost závaží, k_{tz} je tření závaží limitně se blíží k nule, $\dot{x}_Z(t) = v_Z(t)$ a jedná se o rychlost zavěšeného tělesa ve vodorovné poloze a x_K je poloha kočky.

1. stavová rovnice:

$$\dot{x}_Z(t) = v_Z(t) \quad (4.12)$$

2. stavová rovnice:

$$\dot{v}_Z(t) = \ddot{x}_Z(t) = -\frac{g}{l} \cdot x_Z(t) - \frac{k_{tz}}{m_Z} \cdot v_Z(t) + \frac{g}{l} \cdot x_K(t) \quad (4.13)$$

Výstupní veličinou tohoto subsystému je vhodné zvolit úhel vychýlení závěsu φ pro jehož malé hodnoty platí

$$\varphi(t) \approx \sin\varphi(t) = \frac{x_Z(t) - x_K(t)}{l} \quad (4.14)$$

Popis celé soustavy

Shrneme-li jednotlivé popisy subsystémů (4.8), (4.10), (4.12), (4.13), dostaneme rovnici dynamiky.

$$\begin{bmatrix} \dot{v}_M(t) \\ \dot{x}_K(t) \\ \dot{v}_Z(t) \\ \dot{x}_Z(t) \end{bmatrix} = \begin{bmatrix} \frac{-1}{T_{MM}} & 0 & 0 & 0 \\ K_{P\ddot{R}} & 0 & 0 & 0 \\ 0 & \frac{g}{l} & \frac{-k_{tz}}{m_z} & \frac{-g}{l} \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} v_M(t) \\ x_K(t) \\ v_Z(t) \\ x_Z(t) \end{bmatrix} + \begin{bmatrix} \frac{K_{MM}}{T_{MM}} \\ 0 \\ 0 \\ 0 \end{bmatrix} \cdot v_{zM}(t) \quad (4.15)$$

Za výsledné veličiny soustavy volíme všechny 4 stavové veličiny a navíc úhel φ . Rovnice výstupu potom je

$$\begin{bmatrix} v_M(t) \\ x_K(t) \\ v_Z(t) \\ x_Z(t) \\ \varphi(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{-1}{l} & 0 & \frac{1}{l} \end{bmatrix} \cdot \begin{bmatrix} v_M(t) \\ x_K(t) \\ v_Z(t) \\ x_Z(t) \end{bmatrix} \quad (4.16)$$

5.2 Prediktivní regulace

Tato kapitola vychází z [12]

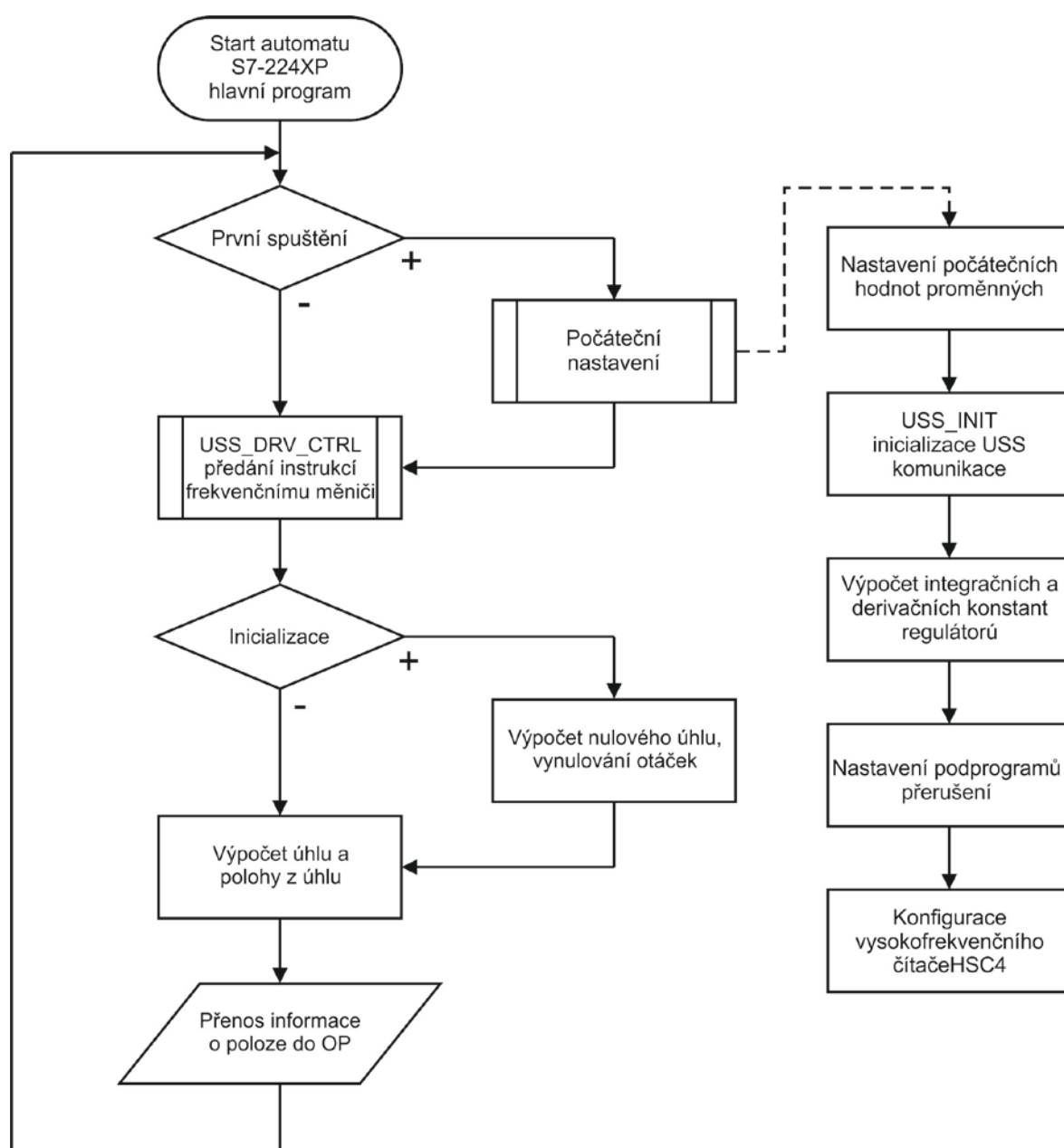
Principem je předpovídání budoucího chování celé soustavy, tedy predikce. Je tedy zapotřebí mít vytvořený model soustavy, na kterém se právě toto budoucí chování bude předpovídat. Toto řízení je diskrétní, protože rozděluje časovou osu na určitý počet konečných časových okamžiků. Pro tyto časové okamžiky se akční zásah optimalizuje následovně. Regulátor vytvoří pro každou množinu časových okamžiků konečnou množinu akčních zásahů, které se následně otestují na modelu soustavy. Z těchto množin akčních zásahů se vybere ta, která má nejlepší vlastnosti regulace na modelu v množině daných časových okamžiků. Proveďte se však pouze první akční zásah z množiny nejlepších akčních zásahů, protože skutečné chování soustavy se od chování modelu může lišit. V dalším časovém okamžiku se celý proces provádí znovu.

Pro tuto regulaci jsou důležité dvě hodnoty. Množina akčních zásahů, kterou regulátor testuje na modelu – Horizont řízení a Horizont predikce, který stanovuje počet časových okamžiků, pro které se testují množiny akčních zásahů. Snad jedinou nevýhodou této regulace je nemožnost řídit proces optimalizace. Lze jej pouze nepřímo ovlivňovat penalizací odchylek jednotlivých stavových veličin. Dále také proto, že se pro velmi malý časový okamžik provádí velké množství výpočtů (Horizontů řízení), je prediktivní metoda regulace velmi náročná na hardwarové nároky. V automatu Simatic S7-224XP by toto řízení kvůli nízkému hardwarovému výkonu ani nebylo možné.

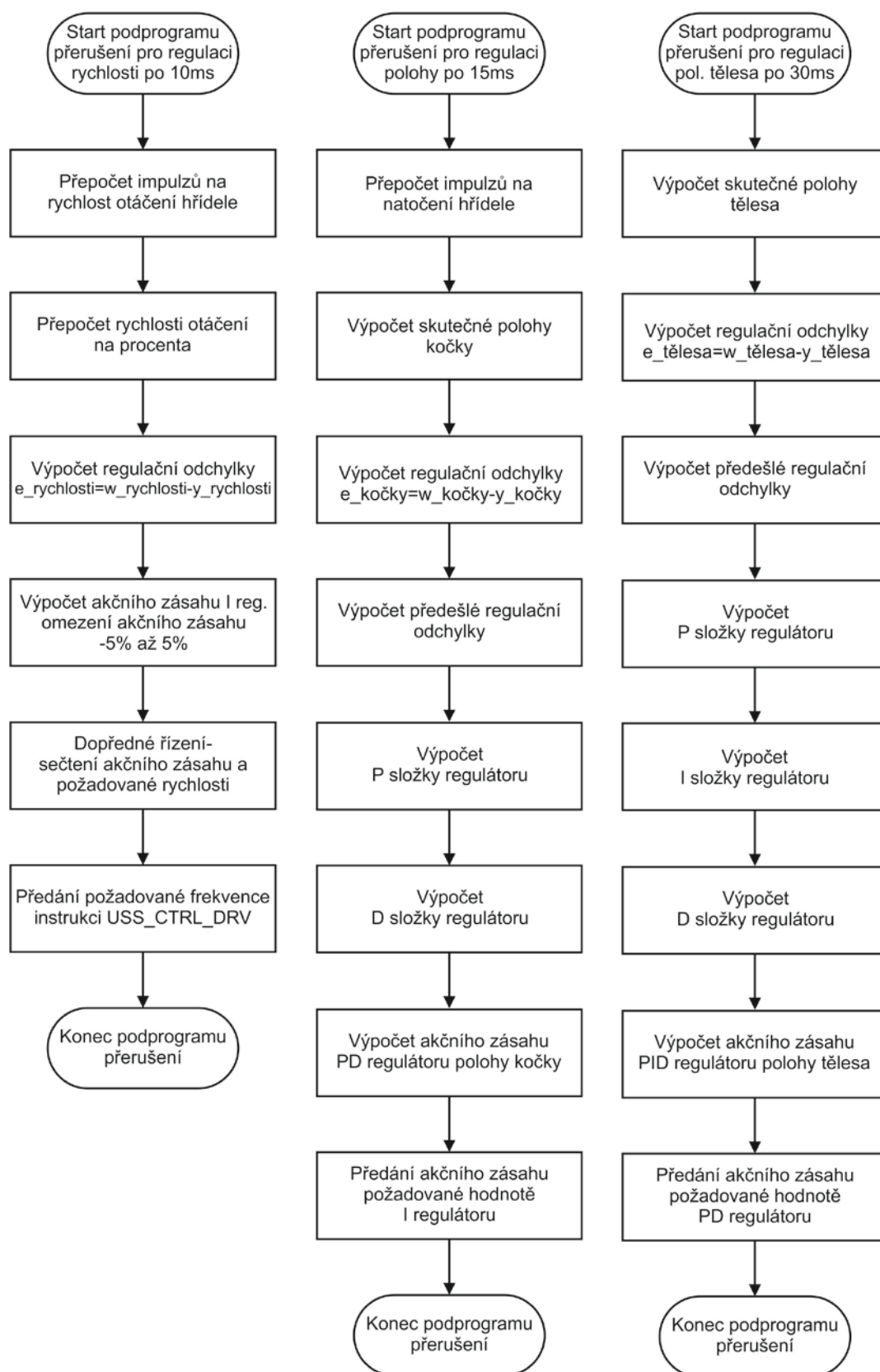
6 Řešení softwarového vybavení automatu

6.1 Vývojový diagram programového vybavení automatu

Na Obr. 6.1 je zobrazena struktura hlavního programu Main s rozhodovacími prvky pro volání podprogramu, strukturu podprogramu a prvky využívajícími zápis pomocí knihoven USS do frekvenčního měniče. Druhý obrázek (Obr. 6.2) znázorňuje strukturu jednotlivých podprogramů přerušení a dobu, po které jsou tyto podprogramy spuštěny. V další kapitole se budu zabývat detailnějším popisem celého programu. Tedy druhou variantou řízení, která vykazovala lepší vlastnosti a kde jsem oproti první variantě vyřadil prostřední smyčku regulace kočky.



Obr. 6.1 Vývojový diagram hlavního programu



Obr. 6.2 Vývojový diagram podprogramů přerušení

6.2 Podrobný popis programu

Aplikace vytvořená v STL kódu se provádí postupně po jednotlivých instrukcích. Hlavní program Main se tedy provádí od první instrukce k poslední. Po dokončení poslední instrukce se vrátí znovu znovu na začátek, a znovu provádí všechny instrukce. Volání celého programu je tedy prováděno cyklicky. V hlavním programu se mohou volat podprogramy, a to vždy, při každém cyklu, nebo v závislosti na nějaké podmínce. Narazí-li program Main na volání určitého podprogramu, odskočí do něj a provádí instrukce opět od první k poslední. Na konci tohoto podprogramu, po dokončení poslední instrukce se proces opět vrátí do hlavního programu a pokračuje dál. Díky tomu se časy zpracování každého cyklu mohou měnit a nebylo by možné provádět některé příkazy v přesně stanovený čas. Pro tyto účely je v prostředí Step 7-microWIN možné vytvořit podprogramy přerušení. Jedná se také o podprogram, který je však spouštěn v přesně stanovených časových intervalech a v závislosti na události, pro kterou se volají.

6.2.1 Hlavní program Main

Z vývojového diagramu (viz. Obr. 6.1) je patrné, že jako první instrukce se provádí volání podprogramu. Tento podprogram má číselné označení SBR0 a pojmenoval jsem si jej „Pocatecni_nastaveni“. Příkaz pro volání podprogramu je „CALL“. Aby bylo zajištěno, že se tento podprogram provede pouze při prvním spuštění automatu, je před tento příkaz předřazen bit „SM0.1“ který je v logické 1 pouze při prvním běhu programu. Ukázka volání podprogramu „Pocatecni_nastaveni“ je na Obr. 6.3.

Network 1 Network Title

Nastavení počátečních hodnot

```
LD      First_Scan_On:SM0.1
CALL    Pocatecni_nastaveni:SBR19
```

Obr. 6.3 Ukázka volání podprogramu „Pocatecni_nastaveni“

Další v pořadí je instrukce USS4_DRV_CTRL, která zapisuje hodnoty do frekvenčního měniče. Instrukce USS4 je blíže popsána v kapitole 2.3.4. Tato instrukce je spouštěna při každém cyklu programu. To je dáno bitem „SM0.0“, který je vždy v logické 1. Při každém zápisu do frekvenčního měniče se pro vizuální kontrolu navíc rozsvítí dioda výstupu „Q0.0“. Tyto výstupy nijak nevyužívám, proto je mohu využít tímto způsobem. Jednotlivé proměnné si v prostředí Step 7-microWIN mohu pro přehlednost pojmenovat a také k nim přidat komentář. Použité proměnné každého networku (obvodu) se pro přehlednost zobrazí v tabulce na konci každého networku. (viz. Obr. 6.4).

Network 2

Instrukce USS4_DRV_CTRL slouží pro řízení aktivního pohonu MicroMaster.

```
LD      Always_On:SM0.0
A       Q0.0
CALL    USS4_DRV_CTRL:SBR5, RunPohonu:V0.1, Zastav:V0.3, V3.0, I0.3,
```

Symbol	Address	Comment
Always_On	SM0.0	Always ON
Error_CTRL	VB13	Cybový byt - USS_CTRL
Poz_ot_motoru	VD1000	Požadované otáčky motoru v %
RunPohonu	V0.1	Zap/Vyp motoru
Smer_otaceni	V0.2	Volba směru otáčení motoru
Speed	VD1004	Rychlost pohonu jako procento plných otáček
Status_CTRL	VW8	Stav - je nezpracovaná hodnota stavového slova
USS4_DRV_CTRL	SBR5	DRV_CTRL// Subroutine call to control a motor drive and g...
Zastav	V0.3	Okamžité zastavení motoru

Obr. 6.4 Ukázka instrukce USS4_DRV_CTRL

V dalším networku (viz. Obr. 6.5) provádím výpočet absolutního úhlu z analogového vstupu, do kterého vstupuje napětí z potenciometru. Tento výpočet se provede pouze při inicializaci, tedy při nastavení výchozí polohy. Aktuální úhel na stupnici potenciometru se zapíše do proměnné. Tento úhel budu v dalším kroku odečítat od aktuálního úhlu. Tím dostanu nulový úhel při svislé poloze závěsu. Další instrukcí vynuluji proměnnou „Počet_skut_ot:VD1050“ Proto je i poloha kočky nulová.

Samotný výpočet úhlu provádím sejmutím hodnoty z analogového vstupu „AIW0“. Velikost této hodnoty je word a proto ji musím nejprve převést na reálnou hodnotu. Tu podělím směrnici přímky (viz. kapitola 0) a dostanu hodnotu úhlu, který ukazuje ručička potenciometru, avšak v převrácené hodnotě. Proto od této hodnoty odečtu maximální úhel naměřitelný na potenciometru a tím dostanu hodnotu, která odpovídá hodnotě na stupnici.

Network 3

inicializační úhel

```
LD      Nulovani:V30.2
MOVR    0.0, uhel_skut:VD1300
MOVW    0, #akuI:LW0
MOVW    AIW0, #akuI:LW0
ITD     #akuI:LW0, #akuD:LD2
DTR     #akuD:LD2, #akuR:LD6
/R      57.576, #akuR:LD6
-R      282.2, #akuR:LD6
MOVR    #akuR:LD6, uhel_skut:VD1300
MOVR    0.0, Pocet_skut_ot:VD1050
R       Nulovani:V30.2, 1
```

Obr. 6.5 Výpočet úhlu z napětí analogového vstupu

V dalším kroku provádím stejný výpočet úhlu, ale od tohoto úhlu navíc odečítám úhel získaný při inicializaci. Proto, nachází-li se těleso v klidu, je tento úhel nulový. V průběhu výpočtu zapisuji jednotlivé mezivýsledky do pomocné proměnné, a až na konci výpočtu ji převedu do finální proměnné, kterou nadále využívám. Toto opatření jsem

zavedl proto, že hlavní program Main je přerušován jednotlivými podprogramy přerušení, které s touto hodnotou pracují. Toto přerušení může nastat například při mezivýpočtu této hodnoty a v proměnné by byl zapsán pouze mezivýsledek a ne skutečná hodnota úhlu.

Pokračováním hlavního programu se vypočítá poloha tělesa z úhlu natočení φ . (viz. Obr. 6.6). Úhel φ z předešlého výpočtu převedu na radiány a podle vzorce

$$x = \sin\varphi \cdot l_z \quad (6.1)$$

převedu na polohu předmětu. Kde l_z je vzdálenost tělesa od osy otáčení potenciometru. Změřením tohoto závěsu jsem dostal hodnotu $l_z = 176\text{mm}$. Tímto program Main končí a cyklus se začne provádět od prvního networku, respektive druhého, protože první se provádí pouze při prvním cyklu.

Network 5

```
přepočet úhlu na vzdálenost
LD      Always_On:SM0.0
MOVR    uhel_nul:VD1304, pol_předm_uhlova_pom:VD1328
*R      3.14159, pol_předm_uhlova_pom:VD1328
/R      180.0, pol_předm_uhlova_pom:VD1328
SIN      pol_předm_uhlova_pom:VD1328, pol_předm_uhlova_pom:VD1328
*R      176.0, pol_předm_uhlova_pom:VD1328
MOVR    pol_předm_uhlova_pom:VD1328, pol_předm_uhlova:VD1308
```

Obr. 6.6 Přepočet úhlu na polohu tělesa

6.2.2 Podprogram počáteční nastavení

Tento podprogram „Pocatecni_nastaveni:SBR0“ je prováděn pouze při zapnutí automatu. Vynulují si zde všechny globální proměnné, inicializuje komunikaci s frekvenčním měničem, a vypočítá integrační a derivační konstanty. První network tedy nuluje proměnné, druhý provádí inicializaci USS4. Ve třetím se provádí výpočet integrační složky (viz. Obr. 6.7), který se se změnami regulované soustavy nijak nemění, a proto jej stačí vypočítat jednou na začátku. Jedná se o podíl mezi dobou vzorkovací periody a integrační časovou konstantou. Do vynulované proměnné „Ci“ vložím hodnotu vzorkovací periody „Ts_0“, tuto hodnotu podělím časovou integrační konstantou „Ti“. Výsledná hodnota se mi vloží do proměnné „Ci“.

Network 3

```
Výpočet integrační složky S regulátoru Ci = (Ts/Ti)
LD      Always_On:SM0.0 // log.1
MOVR    0.0, Ci:VD1158 // vynulujeme promennou Ci
MOVR    Ts_0:VD1190, Ci:VD1158 // do Ci vlozime Ts (Ci=Ts)
/R      Ti:VD1166, Ci:VD1158 // Ci = Ts/Ti
```

Obr. 6.7 Výpočet integrační složky

Dále se provádí výpočet derivační konstanty (viz. Obr. 6.8), jenž je podíl mezi derivační časovou konstantou a dobou vzorkovací periody, tento podíl následně vynásobíme s proporcionálním zesílením. Do vynulované proměnné „Cd“ vložím hodnotu časové derivační konstanty „Td“, tuto hodnotu podělím časovou integrační konstantou „Ts_1“ a vložím do „Cd“. V dalším řádku tuto hodnotu vynásobím hodnotou proporcionálního zesílení Kpp a výslednou hodnotu vložím do „Cd“. Stejným způsobem se

vypočítají i derivační a integrační konstanty pro regulátor polohy tělesa. Liší se jen hodnotami jednotlivých proměnných. Vzorkovací perioda rychlostního regulátoru je stanovena na 10ms a rychlostního na 15ms. Tyto hodnoty proměnných „Ts_0“ a „Ts_1“ jsou uloženy jako konstanty v datovém bloku. Ostatní jsou stanovovány při ladění soustavy a také uloženy v datovém bloku.

Network 4

Výpočet derivační složky pro PD regulátor $C_d = K_{pp} \cdot (T_d / T_s)$

```
LD      Always_On:SM0.0          // log.1
MOVR    0.0, Cd:VD1162           // vynulujeme promennou Cd
MOVR    Td:VD1134, Cd:VD1162     // do Cd vlozime Td (Cd=Td)
/R      Ts_1:VD1154, Cd:VD1162   // Cd = Td/Ts
*R      Kpp:VD1078, Cd:VD1162    // Cd = Kp*(Td/Ts)
```

Obr. 6.8 Výpočet derivační složky

Důležitým krokem v dalším networku (viz. Obr. 6.9) je povolení časových přerušení. Časová perioda spouštění těchto podprogramů je volitelná od 1ms d 255ms. Tento čas zapisují do registrů časových intervalů přerušení SMB34 pro přerušení „INT0“ a SMB35 pro přerušení „INT5“. V těchto podprogramech přerušení probíhají výpočty jednotlivých regulátorů, které popíši níže.

Network 7

Nastavení časového přerušení 0

```
LD      Always_On:SM0.0
MOVB    Time_RR:VB15, Time_0_Intrl:SMB34 // nastavení intervalu preruseni na 10 ms
ATCH    Dopredne_rizeni:INT0, 10
```

Network 8

Nastavení časového přerušení 1

```
LD      Always_On:SM0.0
MOVB    Time_RP:VB10, Time_1_Intrl:SMB35 // nastavení intervalu preruseni na 15 ms
ATCH    Regulace_polohy:INT5, 11
ENI                                           // Povolení všech přerušení.
```

Obr. 6.9 Povolení časových přerušení

6.2.3 Podprogram přerušení pro rychlostní smyčku regulátoru

Vnitřní smyčky bývá zvykem realizovat s vyšší vzorkovací periodou, proto jsem periodu tohoto časového přerušení nastavil na 10ms. Vyzkoušel jsem i kratší periodu, ale vlastnosti regulace se téměř nezměnily a navíc by kratší čas nemusel stačit na výpočet.

Jako první se v tomto podprogramu „INT0“ vypočítá rychlost otáčení motoru tak, že přesuneme aktuální hodnotu vysokorychlostního čítače, tedy počet impulzů z inkrementálního snímače za dobu 10 ms, do proměnné „AktHodCit:VD1146“ a zároveň do pomocné proměnné „Pom2:VD1400“. Od této hodnoty se odečte předchozí hodnota čítače „PredHodCitace:VD1186“. Dále se pak přesune hodnota z „Pom2:VD1400“ do „PredHodCitace:VD1186“, tím získám hodnotu, která bude v dalším kroku použita pro určení rozdílu současného a aktuálního stavu čítače. Takto získaný rozdíl se vynásobí konstantou 6000, tak získám počet impulzů za minutu. Tato hodnota se podělí počtem impulzů na jedno otočení, což je 1024, a dostanu skutečné otáčky za minutu. Ty si v dalším kroku převedu na napětí analogového výstupu, abych mohl velikost otáček odečítat na osciloskopu. Tato část je zobrazena na Obr. 6.10.

Network 1

```

LD      Always_On:SM0.0           //log. 1
MOVD    HC4, AktHodCit:VD1146      // presun stavu citace do VD1146
MOVD    AktHodCit:VD1146, Pom2:VD1400 // stav citace vlozime do LD0
-D      PredHodCitace:VD1186, AktHodCit:VD1146 // od aktualniho stav
MOVD    Pom2:VD1400, PredHodCitace:VD1186 // presuneme LD0 do predch
MOVD    AktHodCit:VD1146, Pocet_pulsu_ekoder:VD1142 // pocet pulsu

LD      Always_On:SM0.0           // log. 1
DTR     AktHodCit:VD1146, aktualni_ot_snimac:VD1028 // Konvertuje ho
*R      6000.0, aktualni_ot_snimac:VD1028 // pocet pulsu za 1 min
/R      1024.0, aktualni_ot_snimac:VD1028 // pocet otoceni za 1 min
MOVR    aktualni_ot_snimac:VD1028, AktOtNeuprav:VD1404 // VD1028 pře

```

Obr. 6.10 Výpočet rychlosti otáčení

V dalším networku (viz. Obr. 6.11) se již provádí výpočet akčního zásahu tak, že si rychlost otáčení, převedenou na procenta vložíme do proměnné „Y_dop“ a požadovanou hodnotu, tedy hodnotu z polohového regulátoru, do „W_dop“. Rozdílem těchto dvou hodnot dostaneme hodnotu regulační odchylky, uloženou v proměnné „E_dop“. V dalším kroku se vypočítá akční zásah tohoto rychlostního regulátoru podle vzorce (4.1) následovně. Regulační odchylka „UIn“, v tomto případě předešlá, se vloží do proměnné UIn_1. Do regulační odchylky „UIn“ se zapíše nula a následně pak hodnota Ci, kterou máme již spočítanou výše v počátečním nastavení. Dále pak tuto hodnotu vynásobíme s regulační odchylkou a přičtu k ní hodnotu předešlého akčního zásahu. Protože integrační složka může růst do nekonečna, je omezena shora i zdola na 5%. Tato hodnota akčního zásahu se na konci tohoto přerušení zapíše do proměnné „Poz_ot_motoru:VD1000“, a v programu Main se pak odešle frekvenčnímu měniči.

```

// Výpočet regulační odchylky e_dop = w_dop - y_dop

LD      Always_On:SM0.0           // log.1
MOVR    0.0, E_dop:VD1412         // vynulujeme e_rych
MOVR    W_rych:VD1178, E_dop:VD1412 // do e_rych vložíme požadovanou hod
-R      Y_dop:VD1408, E_dop:VD1412 // e_dop = w_dop - y_dop

// výpočet integrační složky UIn = UIn_1 + [Ci * e(n)]

LD      Always_On:SM0.0           //minulou integrační složku vložíme
MOVR    UIn:VD1430, UIn_1:VD1434 // vynuluje akční veličinu
MOVR    0.0, UIn:VD1430           //Ci do UIn
MOVR    Ci:VD1158, UIn:VD1430     // Ci * e(n)
*R      E_dop:VD1412, UIn:VD1430 // [UIn = UIn_1 + [Ci * e(n)]]
+R      UIn_1:VD1434, UIn:VD1430 // -----
//MOVR    0.0, UIn:VD1430         // omezení I složky +
LDR>    UIn:VD1430, 5.0           // ...
MOVR    5.0, UIn:VD1430           // omezení I složky -
LDR<    UIn:VD1430, -5.0          // ...
MOVR    -5.0, UIn:VD1430

```

Obr. 6.11 Výpočet akčního zásahu S regulátoru rychlosti

6.2.4 Podprogram přerušení pro polohovou smyčku regulátoru

Tato vnější smyčka má menší prioritu, proto má také delší vzorkovací periodu, tedy 15ms. V tomto podprogramu také odečítám pulzy z inkrementálního snímače, ale nepřevádím je na rychlost ale na natočení hřídele. Hodnotu čítače odečtu od předchozí hodnoty čítače a dostanu počet skutečných impulsů za posledních 15ms. Tuto hodnotu převedu na reálné hodnoty a podělím počtem impulsů snímače na jednu otáčku, tedy 1024. Následnou sumaci impulsů za každé časové přerušení, dostanu skutečnou polohu natočení hřídele motoru. Tyto výpočty jsou zobrazeny na Obr. 6.12.

Network 2

odečítání polohy z enkoderu

```
LD      Always_On:SM0.0           // log. 1
MOVD    HC4, Pulsy:VD1082         // presun stavu citace do VD1082
MOVD    Pulsy:VD1082, AktHodC:VD1260 // pulsy přesunem do LD20
-D      PredHodCit:VD1150, AktHodC:VD1260 // od LD20 odečteme VD1150
MOVD    Pulsy:VD1082, PredHodCit:VD1150 // počet. pulsu vložíme do V
```

Network 3

přepočítání enkoderu na skutečné otáčky

```
LD      Always_On:SM0.0
DTR     AktHodC:VD1260, DwtoR:VD1264 // Konvertuje hodnotu na reálné
/R      1024.0, DwtoR:VD1264         // vydelime realne cislo 1024 a
+R      DwtoR:VD1264, Pocet_skut_ot:VD1050 // Provadi sumaci otacek
```

Obr. 6.12 Výpočet polohy natočení hřídele

V dalším kroku se převádí otáčky na polohu kočky, a to jednoduchým podělením 5, a ukládám ji do proměnné „poloha_kocky:VD1316“. Protože na jednu otáčku se rameno s kočkou přesune o 5mm. K této hodnotě se následně přičte poloha tělesa, a tím získám skutečnou polohu zavěšeného předmětu, uloženou v „poloha_telesa:VD1312“. Tato část je zobrazena na Obr. 6.13.

Network 4

přepočítání poloha - otáčky

```
MOVR    Pocet_skut_ot:VD1050, poloha_kocky:VD1316 //přepočítání skutečn
*R      5.0, poloha_kocky:VD1316

MOVR    poloha_kocky:VD1316, poloha_telesa:VD1312 //přičtení polohy
+R      pol_předm_uhlova:VD1308, poloha_telesa:VD1312
```

Obr. 6.13 Přepočítání otáček na polohu kočky a skutečnou polohu zavěšeného tělesa

Další network již provádí samotný výpočet akčního zásahu PSD regulátoru polohy tělesa. Skutečnou polohu tělesa si tedy vložím do proměnné „Ypol“ a požadovanou polohu tělesa kterou zadává uživatel z operátorského panelu do proměnné „Wpol“. Rozdílem těchto hodnot získávám regulační odchylku polohy tělesa „Epol“. Zároveň si ji uložím do „Epol_1“ jako předešlou hodnotu regulační odchylky. Následně se provádí výpočty jednotlivých složek PSD regulátoru podle vzorce (4.3). Nejprve proporcionální složka. Kde regulační odchylka násobí s proporcionálním zesílením, takto získanou hodnotu vložím do proměnné „UPn“. Integrační složka („U_In“) se provádí podobným způsobem jako u rychlostního regulátoru, který jsem popisoval výše. Derivační složka, se vypočítá

jako rozdíl aktuální regulační odchylky a předešlé regulační odchylky. Tato hodnota se vynásobí konstantou „Cd“ popsanou výše, a vloží do proměnné „UDn“. Na konci přerušení se provede sumace složek „UPn“, „U_In“, „UDn“, omezí se akční zásah shora i zdola 110% a tato suma se zapíše do proměnné akčního zásahu „W_rychl:VD1178“ která vstupuje do regulace rychlosti jako požadovaná hodnota. Tento network je zobrazen na Obr. 6.14.

```
LD      Always_On:SM0.0
MOVR    poloha_telesa:VD1312, Ypol:VD1240
MOVR    pozad_poloha_telesa:VD1320, Wpol:VD1244 // přesun VD1054 do Wpol

// urceni hodnoty e_1 (predchozi regulacni odchylka)
LD      Always_On:SM0.0 // log. 1
MOVR    0.0, Epol_1:VD1198 // vynulování předchozí reg. odch.
MOVR    Epol:VD1194, Epol_1:VD1198 // e_1:= e
MOVR    0.0, Epol:VD1194 // e:= 0.0

//urceni hodnoty e=w-y (regulacni odchylka)
LD      Always_On:SM0.0
-R      Ypol:VD1240, Wpol:VD1244 // x = w-y
MOVR    Wpol:VD1244, Epol:VD1194 // e := x

// výpočet proporcionální složky Upn = Kp * e(n)
LD      Always_On:SM0.0 // log. 1
MOVR    0.0, UPn:VD1248 // vynulujeme akční zásah P složky
MOVR    Epol:VD1194, UPn:VD1248 // Upn:=e
*R      Kpp:VD1078, UPn:VD1248 // Upn = Kp*e

//integracni Ui(n)=Ui(n-1)+[Ci*e(n)]
LD      Always_On:SM0.0
MOVR    U_In:VD1272, U_In1:VD1276 //Ui(n)=Ui(n-1) - minulou integracni
MOVR    0.0, U_In:VD1272 //vynulujeme puvodni Ui
MOVR    Ci2:VD1212, U_In:VD1272 // presuneme integracni sloyku do Ui
*R      E_predmet:VD1216, U_In:VD1272 //vynasobime reg odchylku s intrac
+R      U_In1:VD1276, U_In:VD1272 //Ui(n)=Ui(n-1)+[Ci*e(n)]

// výpočet derivační složky UDn = Cd * [e(n) - e(n-1)]
LD      Always_On:SM0.0 // log. 1
MOVR    0.0, UDn:VD1252 // vynulujeme akční zásah D složky
MOVR    Epol:VD1194, UDn:VD1252 // do UDn vložíme reg. odchylku
-R      Epol_1:VD1198, UDn:VD1252 // od reg. odch. odečtem před. reg.
*R      Cd:VD1162, UDn:VD1252 // rozdíl vynásobíme derivační složk

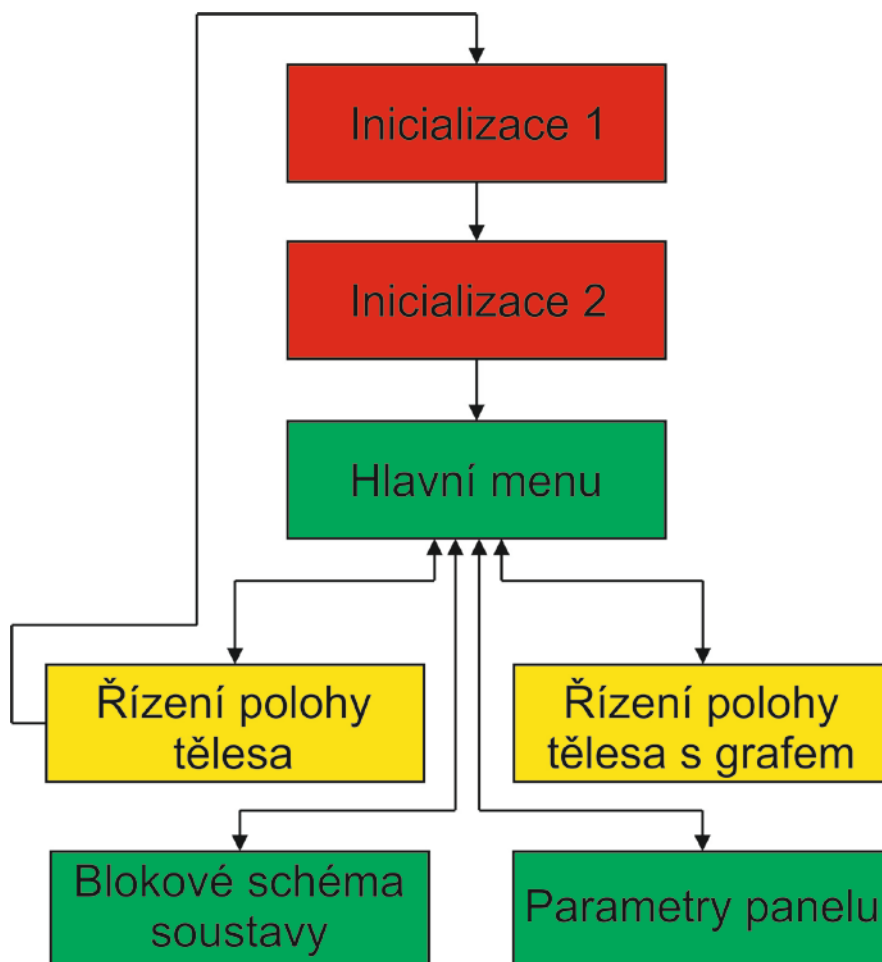
// výpočet akčního zásahu PiD regulátoru Ukn = UPn + UDn
LD      Always_On:SM0.0 // log. 1
MOVR    0.0, UKn:VD1256 // vynulování akčního zásahu PD reg.
MOVR    UPn:VD1248, UKn:VD1256 // přesun UPn do Ukn
+R      UDn:VD1252, UKn:VD1256 // Ukn = UPn + UDn
+R      U_In:VD1272, UKn:VD1256

LDR>    UKn:VD1256, 110.0 // výstupní omezení
MOVR    110.0, UKn:VD1256 // Ukn = 110
LDR<    UKn:VD1256, -110.0 // výstupní omezení
MOVR    -110.0, UKn:VD1256 // Ukn = -110
```

Obr. 6.14 Výpočet akčního zásahu PSD regulátoru polohy tělesa

7 Navržené uživatelské ovládací prostředí

7.1 Struktura uživatelského prostředí

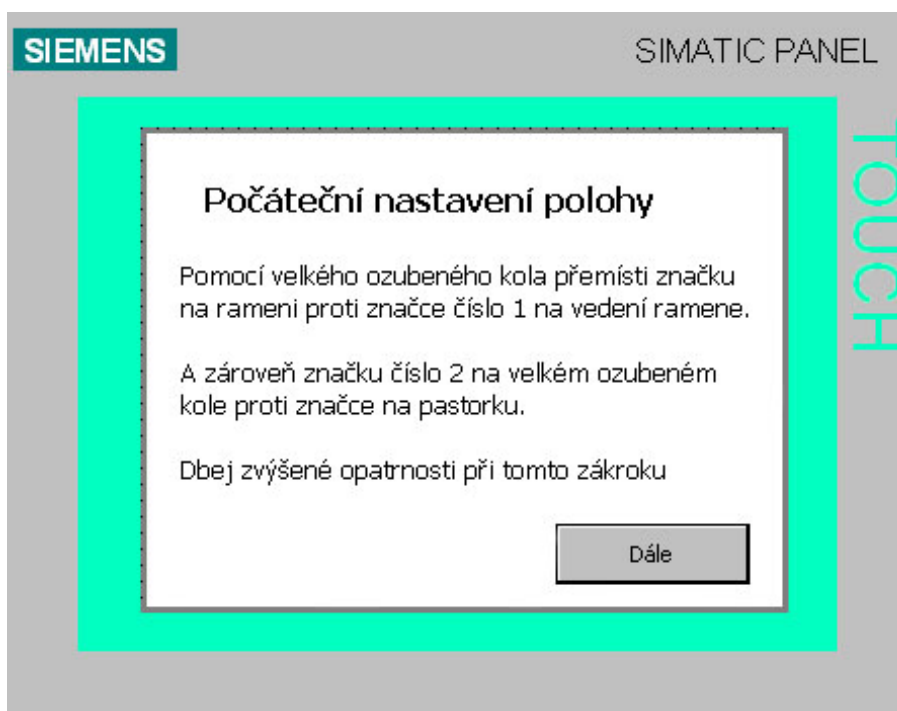


Obr. 7.1 Schéma struktury uživatelského prostředí

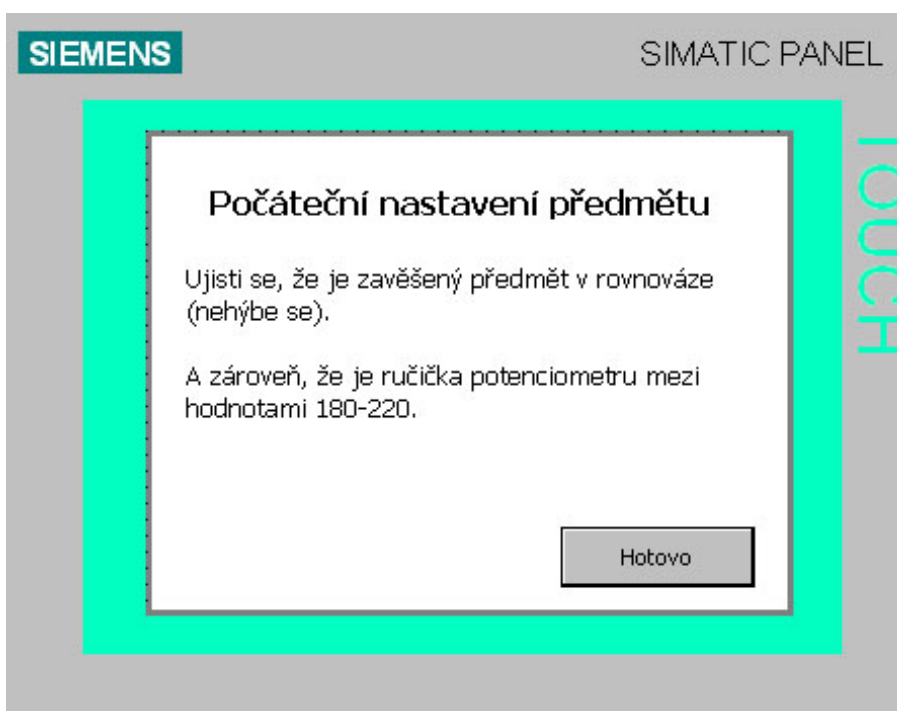
Na Obr. 7.1 je znázorněno schéma uživatelského prostředí, pomocí kterého je uživatel schopen ovládat polohu zavěšeného předmětu. Při spuštění panelu se na chvíli zobrazí systémové menu. Uživatel může vybrat „přenos“ pro nahrávání programu do operátorského panelu nebo nastavení panelu. Když však nic nevybere, panel automaticky zobrazí první obrazovku. V tomto případě obrazovku „Inicializace1“. V tomto okně jsou vypsány instrukce, podle kterých má uživatel postupovat, aby nastavil nulovou polohu kočky. Splní-li uživatel tuto podmínku, stiskne tlačítko „Dále“ a dostane se na obrazovku „Inicializace2“. Zde se po uživateli požaduje, aby zkontroloval, zda je zavěšený předmět svisle a nehoupá se. Po splnění této podmínky stiskne tlačítko „Hotovo“ a dostane se do „Hlavního menu“. Zde si může zobrazit hrubé blokové schéma zapojení celé soustavy, nastavit parametry panelu (kontrast), nebo přejít na řízení zavěšeného předmětu s grafem průběhu otáček a vychýlení či bez. Jednotlivé obrazovky popíše v další kapitole.

7.2 Popis a ukázky jednotlivých obrazovek uživatelského prostředí

Na Obr. 7.2 a Obr. 7.3 jsou zobrazeny jednotlivé inicializační obrazovky. První žádá uživatele, aby nastavil nulovou polohu ramene. Tedy nastavení referenčních bodů na rameni jeřábu a ozubených kolech proti sobě. Ve druhém, aby bylo kyvadlo v klidu a nehoupalo se. Splní-li obě tyto podmínky, dostane se uživatel do „Hlavního menu“.

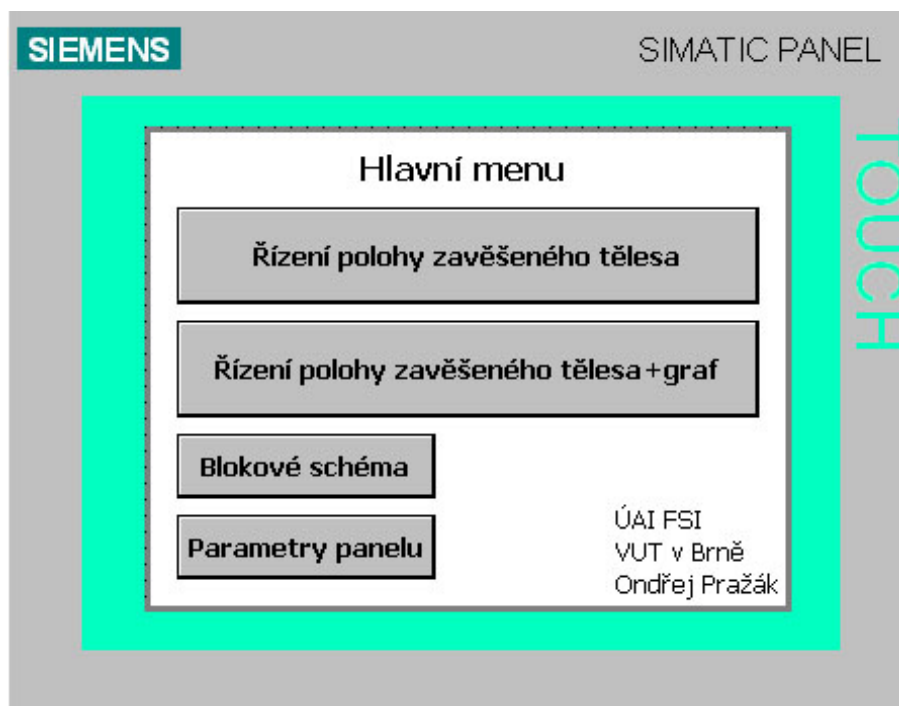


Obr. 7.2 Obrazovka inicializace polohy ramene



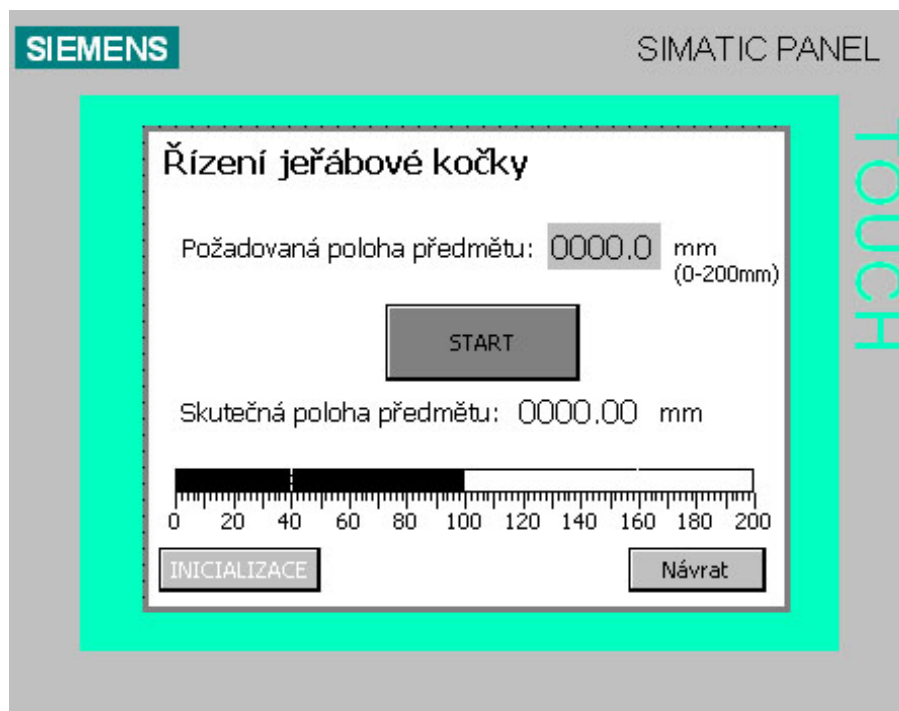
Obr. 7.3 Obrazovka inicializace kyvadla

V hlavním menu (viz. Obr. 7.4) si uživatel může zobrazit blokové schéma zapojení soustavy kliknutím na tlačítko „Blokové schéma“ (viz. Obr. 7.7). Kliknutím na tlačítko „Parametry panelu“ se dostane do obrazovky s jednoduchým ovládáním kontrastu. Pokud uživatel klikne na tlačítko „Řízení polohy zavěšeného tělesa“ dostane se do ovládání polohy tělesa, kde je poloha zobrazována aktuální hodnotou a také na stupnici. (viz. Obr. 7.5). Při kliknutí na tlačítko „Řízení polohy zavěšeného tělesa s grafem“ se uživatel dostane do obrazovky zobrazené na Obr. 7.6, kde může pouze zadávat požadovanou polohu a na grafech se mu zobrazují hodnoty rychlosti motoru a velikosti vychýlení tělesa.



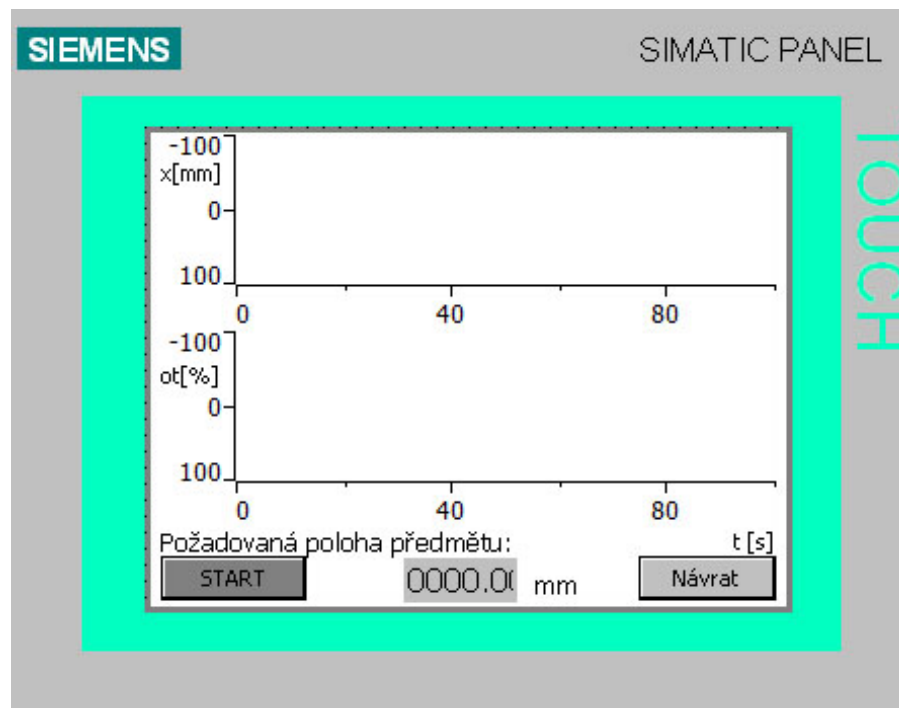
Obr. 7.4 Obrazovka hlavního menu

V okně pro řízení polohy zobrazeného na Obr. 7.5 probíhá samotné řízení polohy tělesa. Kliknutím na číselný údaj „Požadovaná poloha předmětu“ se uživateli zobrazí klávesnice a může zadat polohu tělesa v milimetrech. Hodnota lze zadávat pouze od 0mm do 200mm. Toto omezení je dáno délkou ozubeného hřebene. Stisknutím tlačítka „START“ spustí proces přemísťování, a těleso se regulovaně přemístí na požadovanou polohu. Skutečná poloha je zobrazena pod tlačítkem „START“. Pod číselným údajem skutečné polohy se pro přehlednost ještě zobrazuje skutečná poloha graficky. Pravým tlačítkem „Návrat“ se uživatel vrátí do hlavního menu, a levým tlačítkem „INICIALIZACE“ spustí znovu celý proces inicializace.

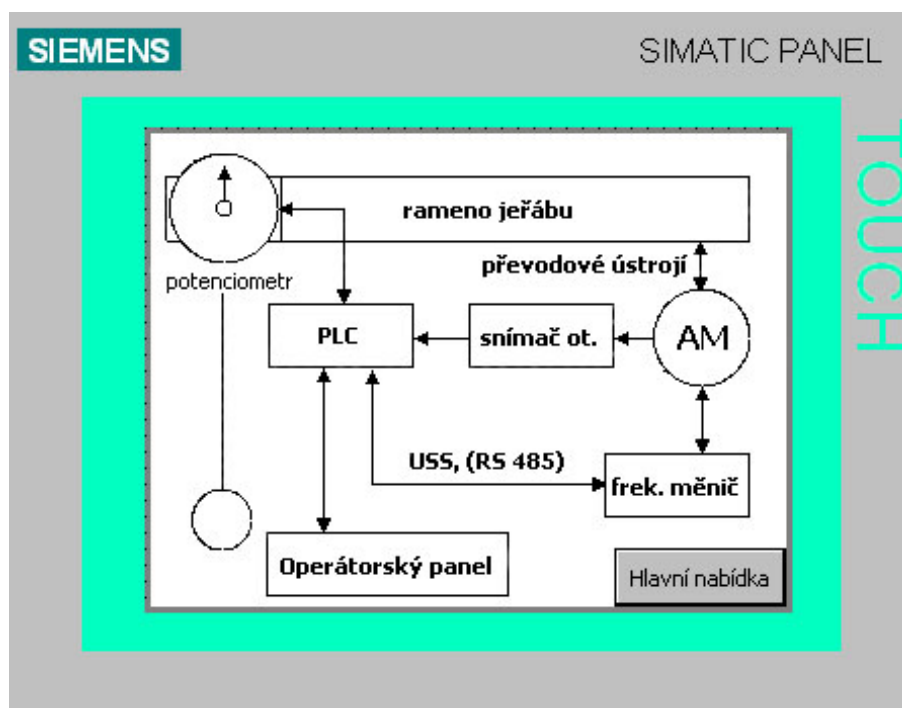


Obr. 7.5 Obrazovka řízení polohy

V okně pro řízení polohy s grafy (viz. Obr. 7.6), může uživatel zadávat požadovanou polohu stejně jako u předchozího okna. Spuštěním se přemístí těleso na požadovanou polohu. V horním grafu se uživateli zobrazí aktuální vychýlení tělesa vůči ose kočky v závislosti na čase. Ve druhém skutečné otáčky motoru v % také v závislosti na čase.



Obr. 7.6 Obrazovka řízení polohy s grafy



Obr. 7.7 Obrazovka s blokovým schématem

8 Laboratorní úloha řízení pohybu při přemísťování zavěšeného předmětu

Pomůcky: Laboratorní přípravek s programovatelným automatem, asynchronním motorem, frekvenčním měničem, operátorským panelem a jeřábovou kočkou.

Úvod

Laboratorní úloha je ukázkou řízení přemísťování zavěšeného předmětu, kde může uživatel zadávat požadovanou polohu předmětu a pomocí motoru a ozubeného převodu se provede akční zásah. Jedná se o kmitavou soustavu, kterou je obtížné regulovat.

1. Seznámení se se soustavou, schéma

Po připojení napájecího napětí se na displeji operátorského panelu zobrazí výzva pro nastavení referenční polohy kočky. Za pomoci velkého ozubeného kola je zapotřebí přesunout značky proti sobě podle instrukcí na panelu. Stisknutím tlačítka **Dále**, se zobrazí instrukce pro nastavení výchozí polohy potenciometru, a především pro kontrolu klidového stavu tělesa. Stisknutím tlačítka **Hotovo**, se uživatel dostane do hlavního menu, kde dvěma menšími tlačítky možné zobrazit blokové schéma zapojení nebo měnit kontrast displeje.

2. Řízení polohy tělesa se skutečnou číselnou hodnotou polohy

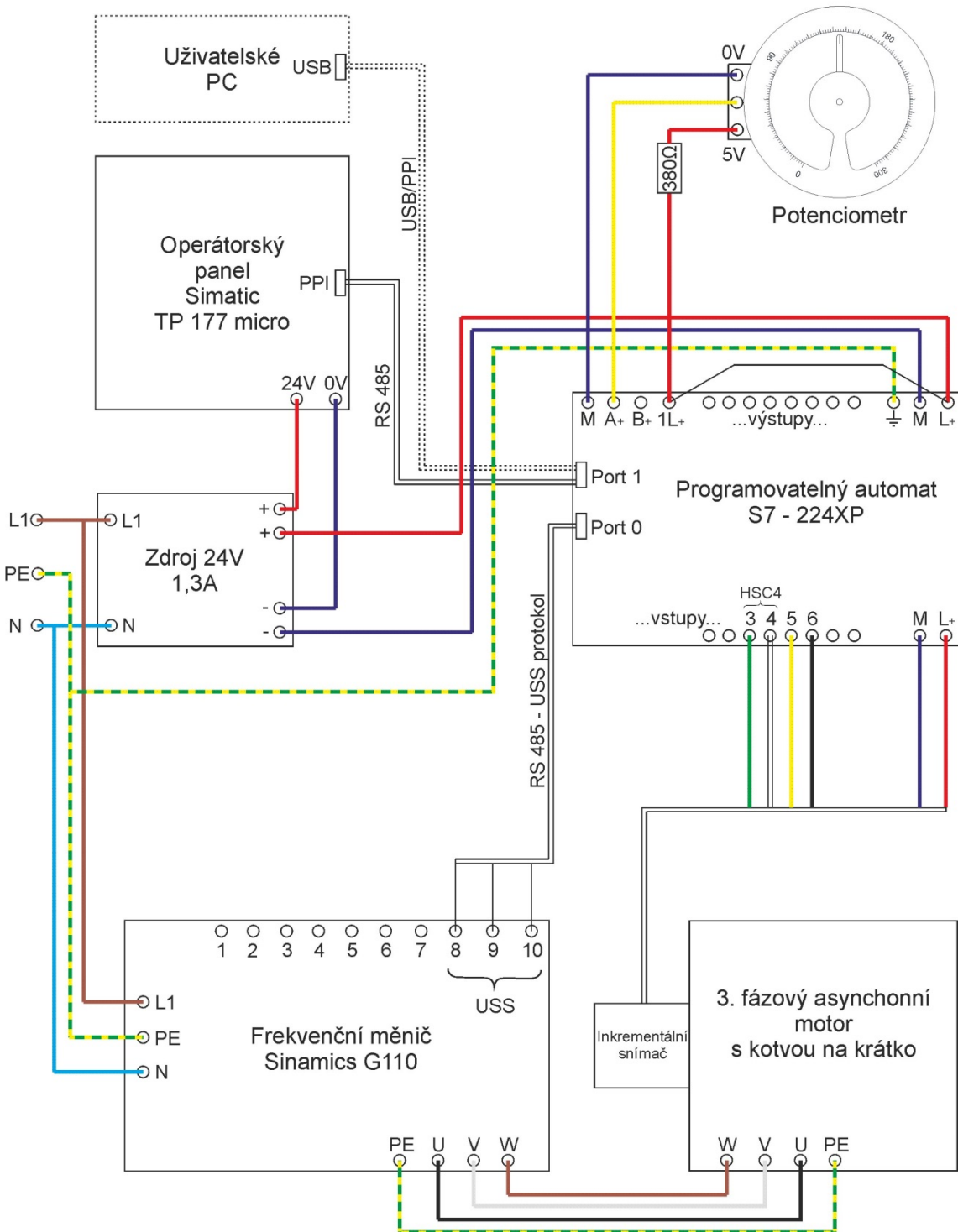
Stisknutím tlačítka **Řízení polohy zavěšeného tělesa** v hlavním menu se dostaneme do okna, kde kliknutím na horní číselný údaj můžeme zadávat požadovanou polohu v mm, ale pouze v rozmezí od 0mm do 200mm. Před stisknutím tlačítka **Start** je vhodné zkontrolovat, zda rameni či ozubenému převodu nepřekáží nějaké předměty. Na dolním číselném údaji se zobrazuje skutečná poloha předmětu, stejně jako na stupnici. Stisknutím tlačítka **INICIALIZACE** se dostaneme znovu do nastavení referenční polohy. Například proto že jsme ji špatně nastavili. Tlačítkem **Zpět** se vrátíme do menu.

3. Řízení polohy tělesa se zobrazením výchylky tělesa a otáček motoru v grafech

Stisknutím tlačítka **Řízení polohy zavěšeného tělesa** v hlavním menu, se nám zobrazí obrazovka s dvěma grafy. Horní zobrazuje vychýlení tělesa vůči své rovnovážné poloze a druhý otáčky motoru v %. V dolní části můžeme opět zadávat požadovanou polohu a ovládat spouštění motoru tlačítkem **Start**. Tlačítkem **Zpět** se dostaneme do hlavního menu.

V tomto okně si také můžeme vyzkoušet chování soustavy při působení vnějších vlivů. Nastavíme požadovanou polohu **100mm** a stiskneme tlačítko **Start**. Jakmile se soustava uklidní, můžeme rukou vychýlit těleso z rovnovážné polohy a sledovat jakým způsobem regulovaná soustava zasáhne.

PŘI PROVÁDĚNÍ TĚCHTO POKUSŮ DBEJTE ZVÝŠENÉ OPATRNOSTI, ABY TĚLESO NEBO JINÝ PŘEDMĚT NEMOHL VNIKNOUT DO OZUBENÉHO PŘEVODU!



Blokové schéma laboratorní soustavy

9 Závěr

Přemísťování zavěšených těles je dnes čím dál častěji využíváno například ve stavebnictví, strojním průmyslu ale i transportu materiálu. Jsou zde kladeny velké požadavky na bezpečnost. Proto musí být obsluha jeřábových zařízení pravidelně vyškolená. Ale i přesto mohou nastat nebezpečné situace vzniklé nepozorností či nezkušeností obsluhy. Například by se těleso svou setrvačností mohlo dostat mimo polohu, na kterou jej chce obsluha umístit a mohlo by způsobit škody či zranění. Proto se některé firmy zabývají vývojem takových zařízení, která by nožná nebezpečí odstranila.

V této práci se zabývám řízením přemísťování zavěšeného tělesa, které by bylo schopné vychýlení vzniklá vnějšími vlivy eliminovat. Pro tuto soustavu jsem navrhl různé varianty řešení, která mají za úkol přemísťovat těleso s co nejmenším kmitáním a překmitem. První varianta spočívala ve vytvoření tří regulačních smyček. Vnitřní smyčka pro regulaci rychlosti, prostřední pro regulaci polohy kočky a vnější pro regulaci polohy zavěšeného tělesa. Takto vzniklý regulátor i přes veškeré snahy však nebylo možné seřadit. Domníval jsem se, že příčina je v zapojení dvou polohových regulátorů za sebou a ve snaze přijít na možnou příčinu vznikla druhá varianta řízení.

Tato varianta se skládá z vnitřní rychlostní smyčky a vnější smyčky, která řídí polohu zavěšeného tělesa. Vnější regulátor je realizován jako PSD, jeho akční zásah je dán rozdílem požadované hodnoty a hodnoty skutečné. Skutečná poloha je součet vychýlení tělesa vůči své rovnovážné poloze a posunutím kočky. Akční zásah této vnější regulační smyčky je přiveden jako žádaná hodnota pro rychlostní regulátor, který je realizován jako sumační. Odečtením skutečné rychlosti vznikne regulační odchylka pro tento regulátor. Z něj je akční zásah posílán pomocí knihoven USS do frekvenčního měniče. Tuto soustavu jsem pomoci Ziegler-Nicholse odladil do takového stavu, kdy vychýlení při přemístění tělesa je minimální s nulovým překmitem.

Protože se soustava nechovala podle mých očekávání, snažil jsem se najít příčinu těchto pesimistických výsledků. Měřením dopravních zpoždění a přenosových rychlostí jsem zjistil, že možná příčina je v převodu analogového signálu z potenciometru na digitální. Tuto hypotézu jsem si ověřil vytvořením jednoduchého programu, který převáděl analogovou hodnotu na digitální a zpět na analogovou a následným měřením jsem zjistil, že převod z analogového vstupu na digitální hodnotu trvá minimálně 125ms. Stejná hodnotu jsem později našel i v katalogovém listu tohoto automatu. Tato doba je příliš velká na to, aby byl tento automat schopen regulovat tuto kmitavou soustavu s lepšími výsledky.

Na závěr jsem vytvořil uživatelské rozhraní pro ovládání celé soustavy. V jedné z variant může uživatel zadávat požadovanou polohu a ta skutečná je zobrazena formou číselného údaje na displeji a také pro lepší orientaci na stupnici zobrazující skutečnou polohu. Ve druhé variantě ovládání může uživatel zadávat polohu a v grafech na displeji dotykového panelu se zobrazují skutečné otáčky motoru v procentech a vychýlení tělesa vůči své rovnovážné poloze. Pro obě tyto varianty byly vytvořeny laboratorní úlohy.

Použitá literatura

- [1] Programovatelný automat S7-200, Systémový manuál. Vydání 06/2004. 534 s.
Dostupný z WWW:
<http://stest1.etnetera.cz/ad/current/content/data_files/automatizacni_systemy/mikr_osystemy/simatic_s7200/manual_s7_200_2004_cz.pdf>
- [2] Siemens AG Industry Sector 2008: [online]. 10.05.2013. Dostupný z WWW:
<https://eb.automation.siemens.com/collaterals/files/04/jpg/P_ST70_XX_02562i.jpg>
- [3] Vrtal, V., Laboratorní ukázka řízení asynchronního motoru. Diplomová práce VUT v Brně, FSI-ÚAI, 2010
- [4] Siemens AG Industry Sector 2008: [online]. 10.05.2013. Dostupný z WWW: <<http://www.automation.siemens.com/mcems/human-machine-interface/en/operator-interfaces/micro-panels/simatic-tp-177micro/PublishingImages/simatic-tp177-micro-both.jpg>>
- [5] SIMATIC HMI Device TP 177A, TP 177B, OP 177B (WinCC flexible) Operating Instructions. Vydání 08/2008. Dostupný z WWW:
<http://cache.automation.siemens.com/dnl/jQ/jQ0MTc1AAAA_21084461_HB/hmi_tp177a_tp177b_op177b_operating_instructions_en_US_en-US.pdf>
- [6] Eltk: [online]. 10.05.2013. Dostupný z WWW: <<http://eltk.ru/wp-content/uploads/2012/04/Sinamics-G1101.jpg>>
- [7] SINAMICS G110, Návod k obsluze,-stručný. Vydání 11/2004. Dostupný z WWW:
<http://stest1.etnetera.cz/ad/current/content/data_files/technika_pohonu/menice/stri_dave_menice/nizkonapetove_menice/sinamics_g110/_manualy/opic_sinamics_g110_11-2004_cz.pdf>
- [8] Wikipedie, asynchronní motor: [online]. 11.05.2013. Dostupný z WWW:
<http://cs.wikipedia.org/wiki/Asynchronn%C3%AD_motor>
- [9] Kovoterm: [online]. 10.05.2013. Dostupný z WWW:
<http://www.kovoterm.eu/media/images/obrazek_11a7-1.jpg>
- [10] Katalog KO2, Nízkonapěťové motory. Dostupný z WWW:
<http://stest1.etnetera.cz/ad/current/content/data_files/katalogy/k02/cat_k02_2010-04_cz.pdf>
- [11] Siemens71: [online]. 12.05.2013. Dostupný z WWW:
<http://siemens71.ru/images/Content/P_D081_XX_00071.JPG>
- [12] Richter, J., Řízení pohybu při přemísťování zavěšeného předmětu. Diplomová práce VUT v Brně, FSI-ÚAI, 2010
- [13] ŠVARC, I., ŠEDA, M., VÍTEČKOVÁ, M.: *Automatické řízení*. Brno: Cerm, 2007. ISBN 978-80-214-3491-2.
- [14] Konzultace s firmou: ADTEC s.r.o., Vídeňská 117a, 61900 Brno. 02.05.2013.

Seznam obrázků

Obr. 2.1 PLC SIMATICS S7 224 XP [2].....	13
Obr. 2.2 Zapojení RS 485 se zakončovacími odpory [3]	13
Obr. 2.3 Popis jednotlivých pinů konektoru RS 485 [1]	14
Obr. 2.4 Uživatelské prostředí Step 7-MicroWIN	15
Obr. 2.5 Ukázka kódu vytvořeného v editoru STL	16
Obr. 2.6 Operátorský panel SIMATIC TP 177 micro [4]	17
Obr. 2.7 Vývojové prostředí WinCC flexible Micro.....	18
Obr. 2.8 Frekvenční měnič SINAMICS G110 [6]	19
Obr. 2.9 Schéma vnitřního zapojení frekvenčního měniče [7].....	21
Obr. 2.10 Třífázový asynchronní motor SIEMENS [9]	24
Obr. 2.11 Štítek asynchronního motoru SIEMENS	24
Obr. 2.12 Impulzní snímač otáček [11]	25
Obr. 2.13 Závislost číselných údajů na úhlu natočení potenciometru.....	26
Obr. 3.1 Blokové schéma laboratorní soustavy	28
Obr. 3.2 Schéma mechanismu kočky a závěsu	29
Obr. 3.3 Ukázka celého laboratorního setu	30
Obr. 4.1 Schéma první varianty řízení	31
Obr. 4.2 Schéma druhé varianty řízení (použito)	34
Obr. 4.3 Časový průběh otáček motoru a polohy tělesa.....	34
Obr. 4.4 Časový průběh rychlosti motoru	35
Obr. 4.5 Časové průběhy vychýlení závěsu a rychlosti motoru.....	36
Obr. 4.6 Detail časového průběhu vychýlení závěsu a rychlosti motoru	36
Obr. 4.7 Skutečný a převedený časový průběh napětí potenciometru	37
Obr. 5.1 Blokové schéma stavové soustavy	38
Obr. 6.1 Vývojový diagram hlavního programu	41
Obr. 6.2 Vývojový diagram podprogramů přerušení	42
Obr. 6.3 Ukázka volání podprogramu „Pocatecni_nastaveni“	43
Obr. 6.4 Ukázka instrukce USS4_DRV_CTRL.....	44
Obr. 6.5 Výpočet úhlu z napětí analogového vstupu	44
Obr. 6.6 Přepočet úhlu na polohu tělesa.....	45
Obr. 6.7 Výpočet integrační složky.....	45

Obr. 6.8 Výpočet derivační složky	46
Obr. 6.9 Povolení časových přerušení	46
Obr. 6.10 Výpočet rychlosti otáčení	47
Obr. 6.11 Výpočet akčního zásahu S regulátoru rychlosti	47
Obr. 6.12 Výpočet polohy natočení hřídele	48
Obr. 6.13 Přepočet otáček na polohu kočky a skutečnou polohu zavěšeného tělesa .	48
Obr. 6.14 Výpočet akčního zásahu PSD regulátoru polohy tělesa	49
Obr. 7.1 Schéma struktury uživatelského prostředí	50
Obr. 7.2 Obrazovka inicializace polohy ramene	51
Obr. 7.3 Obrazovka inicializace kyvadla	51
Obr. 7.4 Obrazovka hlavního menu	52
Obr. 7.5 Obrazovka řízení polohy	53
Obr. 7.6 Obrazovka řízení polohy s grafy	53
Obr. 7.7 Obrazovka s blokovým schématem	54